NEC

User's Manual

Low-Voltage Starter Kit

For Motor Control

V850ES/IK1

Document No. U19143EE1V0UM00
Date published May 2008

Printed in Germany

Legal Notes

- **The information in this document is current as of November, 2007. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**

- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.

- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.

- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

- The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime

systems, safety equipment and medical equipment (not specifically designed for life support).
"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

Notes for CMOS Devices

1. **VOLTAGE APPLICATION WAVEFORM AT INPUT PIN**
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

2. **HANDLING OF UNUSED INPUT PINS**
   Unconnected CMOS device inputs can result in malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

3. **PRECAUTION AGAINST ESD**
   A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and to quickly dissipate it should it occur. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

4. **STATUS BEFORE INITIALIZATION**
   Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

5. **POWER ON/OFF SEQUENCE**
   In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be

judged separately for each device and according to related specifications governing the device.

6. **INPUT OF SIGNAL DURING POWER OFF STATE**
Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives anddistributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and otherlegal issues may also vary from country to country.

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668, Japan
Tel: 044 4355111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554,
U.S.A.
Tel: 408 5886000
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211 65030
http://www.eu.necel.com/

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908 691133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01 30675800

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 6387200

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02 667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven,
The Netherlands
Tel: 040 2654010

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27
ZhiChunLu Haidian District,
Beijing 100083, P.R.China
Tel: 010 82351155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2511-2512, Bank of China
Tower,
200 Yincheng Road Central,
Pudong New Area,
Shanghai 200120, P.R. China
Tel: 021 58885400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886 9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R.O.C.
Tel: 02 27192377

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253 8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku, Seoul,
135-080, Korea Tel: 02-558-3737
http://www.kr.necel.com/

Preface

Readers This manual is intended for users who want to understand the functions of the low voltage starter kit for motor control.

Purpose This manual presents the hardware manual of the low voltage starter kit for motor control.

Organization This system specification describes the following sections:

- Kit Contents
- Hardware Setup
- Standalone Operation
- Drive and Motor Protection
- Software Setup
- Using the IAR C Compiler and Source Code Debugger
- Download and Debug the Code
- GUI Operation

Legend Symbols and notation are used as follows:

- Weight in data notation: Left is high order column, right is low order column
- Active low notation: $\overline{xxx}$ (pin or signal name is over-scored) or /xxx (slash before signal name)
- Memory map address: High order at high stage and low order at low stage

Note Additional remark or tip

Caution Item deserving extra attention

Numeric Notation

- Binary: xxxx or xxxB
- Decimal: xxxx
- Hexadecimal: xxxxH or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity):

- K (kilo): 210 = 1024
- M (mega): 220 = 10242 = 1,048,576
- G (giga): 230 = 10243 = 1,073,741,824

Table of Contents

Chapter 1 Introduction 9

Chapter 2 Kit Contents 10

Chapter 3 Hardware Setup 11

Chapter 4 Standalone Operation 13

Chapter 5 Drive and Motor Protection 15

Chapter 6 GUI Operation 16

Chapter 7 Programming the Microcontroller 19

7.1 Progamming with the Microcontroller Board disconnected 20

7.2 Progamming with the Microcontroller Board connected 25

Chapter 8 Using the IAR Embedded Workbench 26

8.1 Software Installation 26

8.2 MC-LVKIT-IK1 Setup for On-chip Debugging 26

8.3 IAR Embedded Workbench 29

8.4 Compiler Options 34

8.5 Assembler Options 37

8.6 Linker Options 39

8.7 Integrated Debugger Selection 42

8.8 Workspace and Project Setup if the example is not compatible with the installed IAR workbench 44

8.9 Build / Rebuild the Project 44

8.10 Debugging 45

Chapter 1 Introduction

The low-voltage starter kit for motor control (MC-LVKIT-IK1) is a complete 3-phase motor control evaluation system for NEC Electronics' microcontroller application-specific standard products (ASSP's) for motor control.

The kit contains all necessary hardware and software to quickly set up and run a low-voltage brushless DC (BLDC) motor .

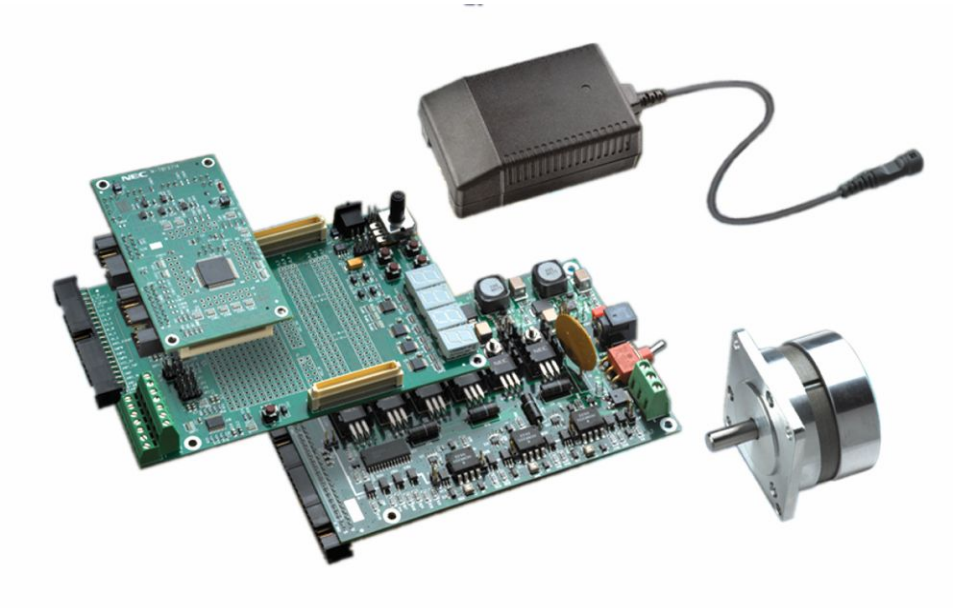

Figure 1-1 Motor Control Starter Kit

Chapter 2 Kit Contents

- MC-CPU-IK1 microcontroller board containing the µPD70F3329 ASSP microcontroller
- MC-I/O interface board. Connects the microcontroller board to the power module
- Low-voltage power module (MC-LV_INVERTER) containing the inverter MOSFET's control and interface circuitry.
- BLDC motor
  - Pitman N2311 12V motor or
  - Ametek 150093-50 24V motor
- FW7362/15 DC power supply
- 20 MHz oscillator (For use with an On Chip Debug Unit)

For information about the electrical characteristics and hardware functions of the µPD70F3329 microcontroller, refer to **µPD70F3329 User's Manual** (U16910EJ3V0UD00).

Chapter 3 Hardware Setup

The kit can be purchased as one unit with all three boards connected as shown below: Replacement board can be ordered separately.

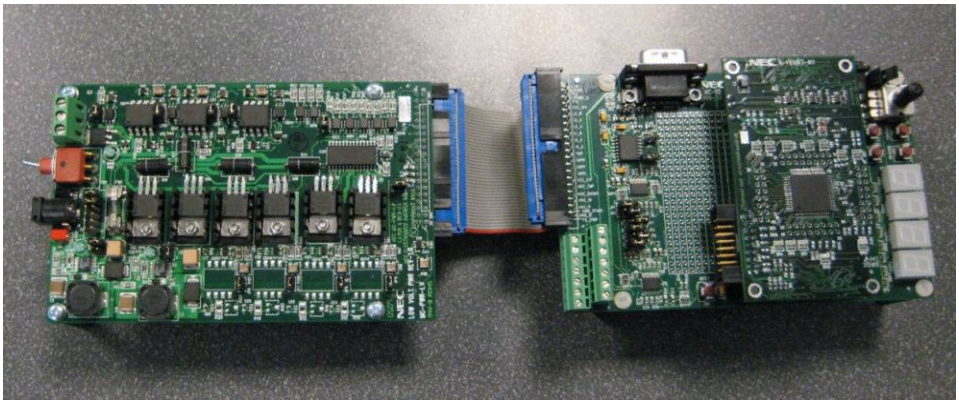

Figure 3-1 MC-LVKIT-IK1 Kit Configuration

To attach the motor, connect the phase U, V and W terminals to the J3 connector block on the MC-PWR-LV power board and the Hall sensor terminals to the J5 connector block on the MC-IO control board.

Table 3-1 Motor Connections

Motor Terminals			I/O Board (MC-IO)	Power Board (MV-PWR-LV)
Connection	Pitman	Ametek		
Phase U	Beige	Black	—	J3 - 1
Phase V	Red	White	—	J3 - 2
Phase W	Orange	Red	—	J3 - 3
Hall sensor 1	Grey	Blue	J5 - 13	—
Hall sensor 2	Blue	Green	J5 - 14	—
Hall sensor 3	White	Yellow	J5 - 7	—
Hall sensor 5VDC	Purple	Purple	J5 - 6	—
Hall sensor GND	Black	Orange	J5 - 5	—

Motor terminal connections are shown in *Figure 3-2* and *Figure 3-3* .

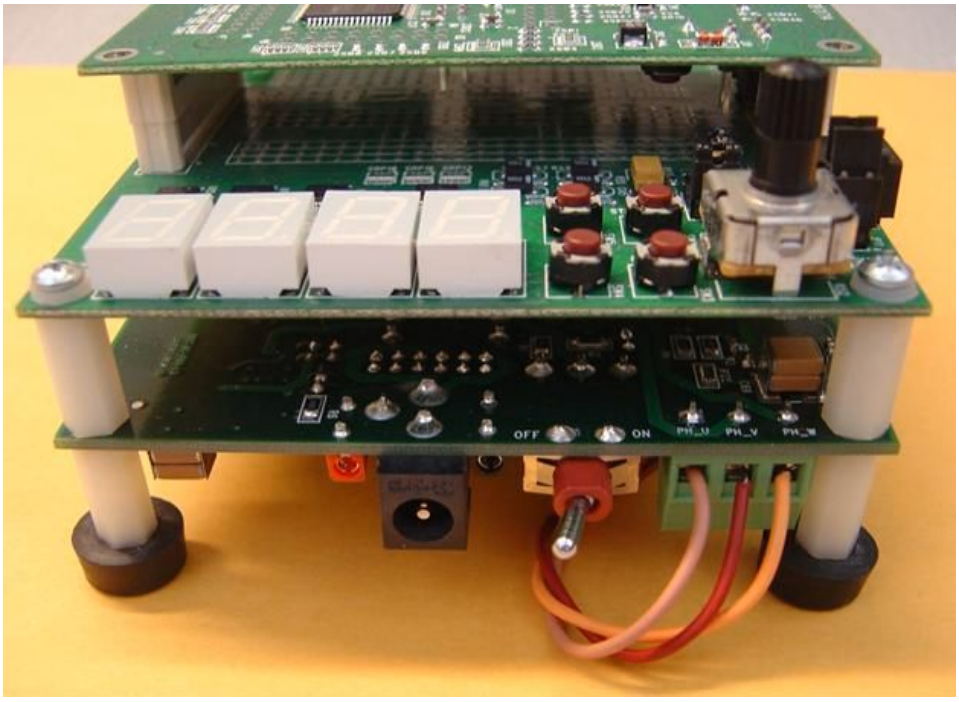

Figure 3-2 Motor Phase Connections

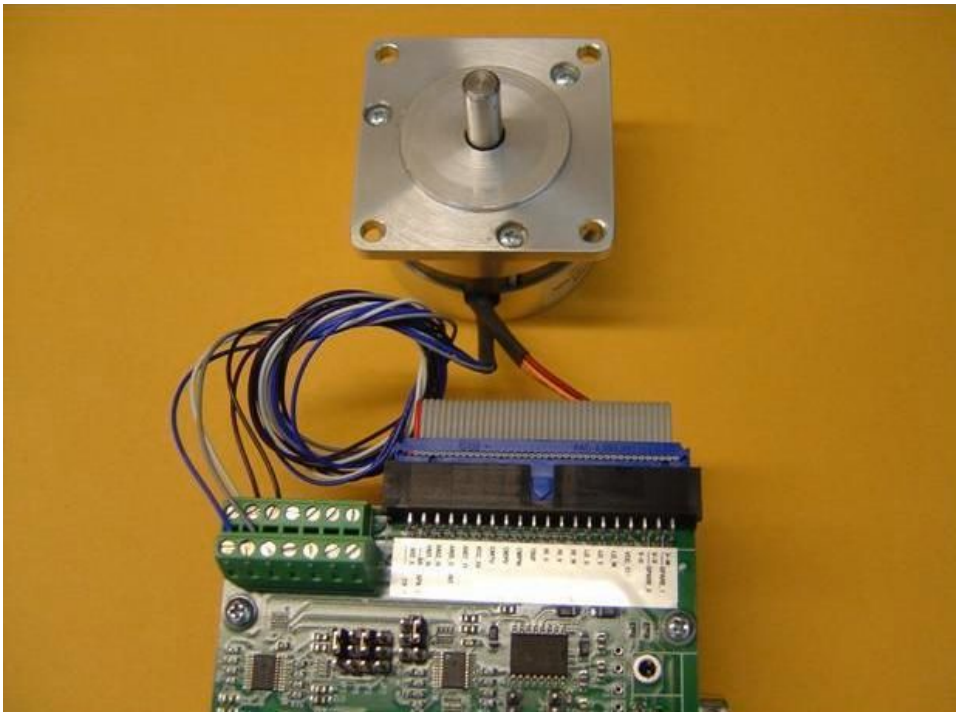

Figure 3-3 Hall Sensor Connections

Figure 3-2 and *Figure 3-3* - Pitman Motor shown for reference.
See *Table 3-1* for equivalent Ametek motor connections.

Chapter 4 Standalone Operation

For Stand alone use (i.e. without the GUI), please check that the following jumpers and links are as defined below

Microcontroller CPU Board

- SB6 Shorted
- JP1 Shorted
- JP2 Open
- JP23 Open
- SB1, 2, 3, 4, 5 All shorted

I/O Board

- JP1 Pins 2 - 3 shorted
- JP2 Pins 2 - 3 shorted
- JP3 Pins 2 - 3 shorted
- JP11 Pins 1 - 2 and 3 - 4 shorted
- JP5 Pins 1 - 2 and 3 - 4 shorted

Low Voltage Power Board (Using 15 V power supply module supplied)

- SB24 Shorted
- SB25 Open
- JP2 Shorted
- J2 Pins 5 and 6 shorted, pins 7 and 8 shorted, all others open

Example software to run the motor is pre-programmed into the microcontroller's flash memory. After the motor is connected, the program is ready to run the motor as soon as the 15V DC power supply is plugged into J6 of the MC-IO board and power switch SW1 on the MC-PWR-LV power module is turned ON.

When the kit is powered up or reset, the LED displays "**SELF**", indicating that the kit is in standalone mode and you can use the pushbuttons and potentiometer on the MC-IO board to control the motor.

Figure 4-1 Stand Alone Operation

Three seconds after power up, the LED displays the current (start-up) set speed. (The speed set depends on the potentiometer position).

Figure 4-2 Initial Speed Display (Pot = 0)

After power up in standalone mode, the motor can be operated as follows:

1. Press the **START/STOP** button to run the motor.
2. Turn the potentiometer clockwise to increase the speed of the motor or counter clockwise to decrease the speed of the motor.

The maximum speed of the Pittman BLDC motor is set to 6000 rpm.

Note The maximum speed of the Ametek motor when driven from the 15V DC power supply is 4000 rpm. Higher speeds are possible using an external 24V power supply (minimum 2A). To remain compatible with both motors the software is set with a maximum speed setting for the Ametek motor to 6000 rpm. Using the external power supply higher speeds may be achieved, however the software will need to be modified. To use an external power supply set the following jumpers on the power board and connect the external power supply to connector J4 on the power board.

- **Jumper J2**
  - Pins 1 and 2 shorted
  - Pins 3 and 4 shorted
  - All others to Open

3. The LED display will show either the actual motor speed calculated from the Hall sensor interrupts, or the target speed setting as set by the potentiometer. The display can be toggled between actual and target speed by use of the **MODE** button. Please note that the function of the mode button is “latched” so that the display will stay set to the selected mode. To change to the alternate display setting, just press the **MODE** button again.
4. Press the **FORWARD** or **REVERSE** button to change the rotation direction. A variable delay is implemented in the software to ensure that the motor stops before the direction is reversed. (This is especially necessary at high speeds)
   FORWARD will run the motor to in a Clockwise direction
   REVERSE will run the motor in a Anti-clockwise direction

Notes

1. As described above, a delay has been built into the control software to allow time for the motor to stop rotating if the direction is reversed. This avoids an error condition occurring due to timeout or over-current detection. If for any reason the motor stops operating due to an error condition (i.e. Hall sensor, over current etc) the software is designed to reset automatically. The display will show the error message for a few seconds and then sequence back through the start procedure. To start the motor press the **START** button as before. If the fault condition persists, check the connections, power etc and if necessary restart the system by pressing the **RESET** button on the MC-IO board
2. To control the motor from the Graphical Interface (GUI) from your PC, please refer to "*section 6*" .

Chapter 5 Drive and Motor Protection

The starter kit and the motor are protected against unexpected events such as overload, motor stall and malfunction of the Hall sensors. If such faults are detected, the motor stops rotating and the fault conditions are displayed on the seven-segment LED. For details on the protection functions implemented in hardware, consult the user's manual for MC-PWR-LV low-voltage power module. The sample code software also has built-in fault detection algorithms as an extra measure of protection. Consult the software manual for details.

In standalone ("SELF") mode, the LED displays the following fault conditions:

- Motor over-current: "O – C – "
- Motor stall fault: "– – – –"
- Hall sensor fault: "H A L L"

In the Graphical Interface (GUI) mode, the LED display will show "**PC**" all the time and the GUI will display all fault conditions. Refer to "*section 6*" for more information.

Chapter 6 GUI Operation

To operate the motor from the PC GUI, the Hex file for the motor control software including the GUI interface, will need to be programmed into the Flash memory of the µPD70F3329 microcontroller. The complete IAR embedded Workbench project can be downloaded from the *Motor Control starter kit web site* .

To program the "*hex*" file (*BLDC_IK1_GUI.hex*) into the microcontroller please refer to "*section 7*" .

Once the Graphical Interface program has been programmed ensure that the jumpers on the microcontroller board are returned to the settings described in "*chapter 4*" above.

It is necessary to locate the following files on to the PC in order to connect to the starter kit. The two programs should always be located together in the same directory.

- MotorPanel.exe (GUI Application)
- Motor.inf (set up file)

To operate the PC GUI a “Female-Female” RS232 serial cable (with crossover) is required. (Not supplied).

Connect the PC serial port to the starter kit J9 DB-9 RS232 connector with an RS232 serial cable.

The pin connection of the RS232 cable needs to be as follows (crossover):

Table 6-1 GUI RS232 Cable Connections

Signal Name	PC Connection	Starter Kit connection
Rx Signal Detect (Not Used)	Pin 1	Pin 1
Rx Data	Pin 2	Pin 3
Tx Data	Pin 3	Pin 2
DTE Ready	Pin 4	Pin 4
Signal Ground	Pin 5	Pin 5
DCE Ready	Pin 6	Pin 6
RTS	Pin 7	Pin 8
CTS	Pin 8	Pin 7
Ring Indicator (Not used)	Pin 9	Pin 9

To launch the communication program Run the "MotorPanel.exe" application program and the following window should open.

1. Select the appropriate COM port on your PC and click OK. (Note: It is possible to use the GUI program with a USB to serial RS232 adapter using an available USB port on the PC.)

Figure 6-1 PC COM Port Selection

The GUI will launch if the connection is established:

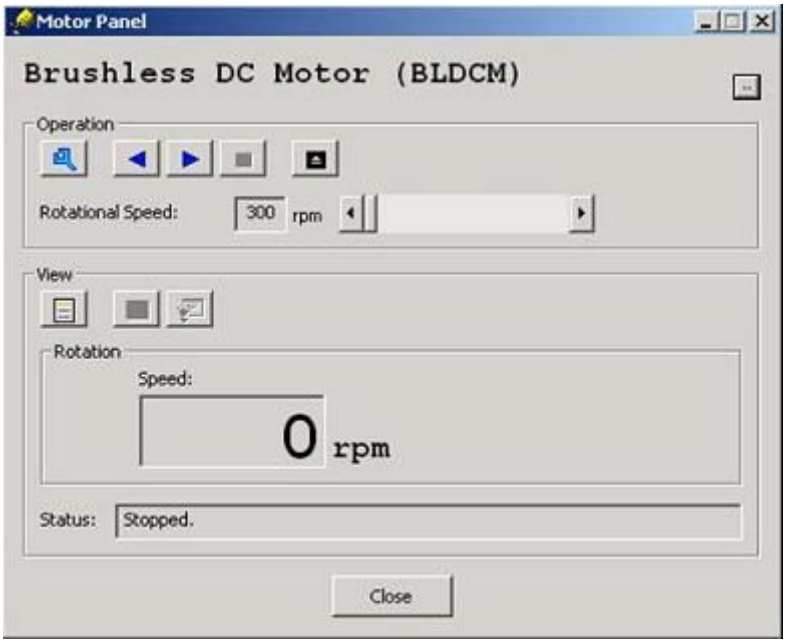

Figure 6-2 BLDC Motor Control GUI

2. To operate the motor, use the controls in the GUI window. The user has the same controls as described for the “Stand Alone” mode (Start / Stop, Clockwise / Anti-Clockwise, Speed increase / decrease)

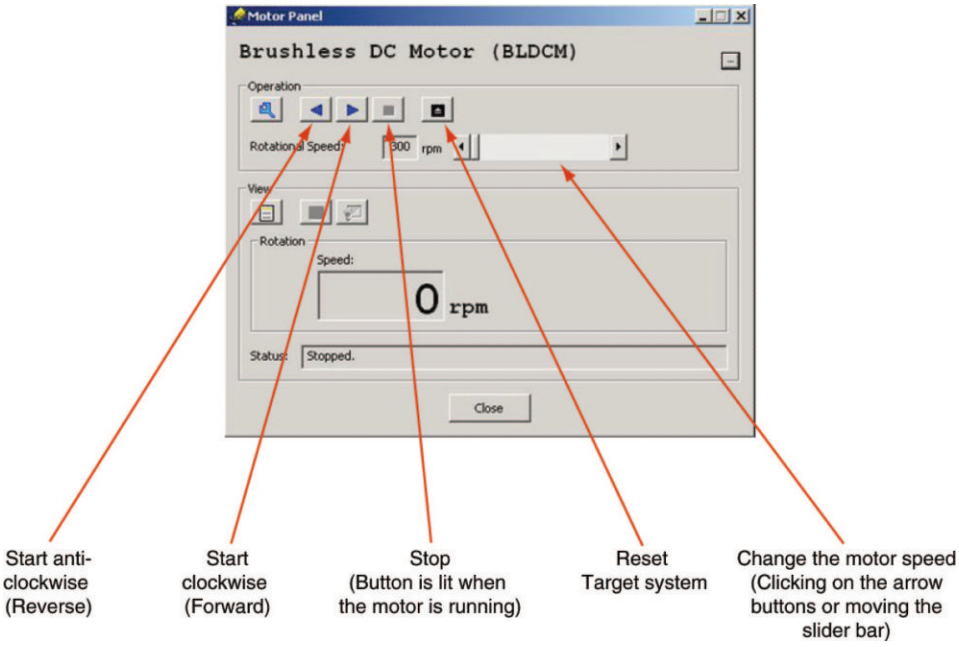

Figure 6-3 BLDC Motor Control GUI Functions

3. The PID parameters can be changed from the GUI interface, so that fine tuning of the control loop is possible. Please note that changes can only be made when the motor is stopped.
To change the PID parameters click on the tuning key symbol as shown below:

Figure 6-4 Launching PID Parameter Window

The tuning window will open allowing the user to change the values of P, I or D constants: To save the new values and update the application in the starter kit, click on the **OK** button.

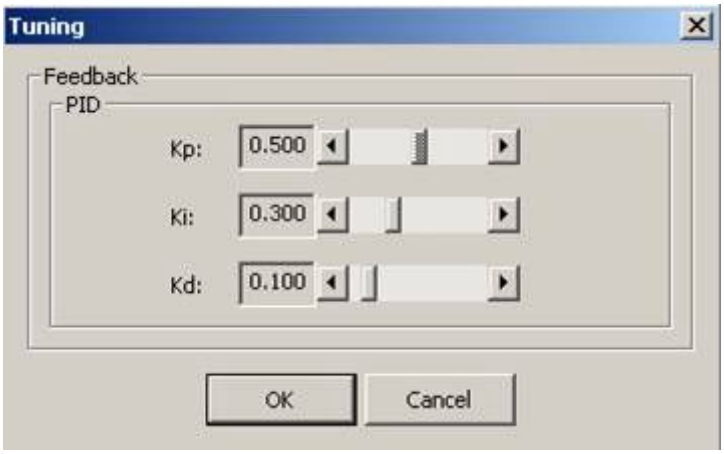

Figure 6-5 PID Parameter Window

Note The new parameter values will be downloaded to the starter kit into the internal RAM only.

To make the changes permanent, the user will have to modify the definitions in the original IAR project file (**control.h**) and then rebuild the project. This requires a licensed version of either a full IAR Embedded Workbench version or a Kick-start IAR Embedded Workbench version.

The three PID values that require changing are

#define KP_DEF

#define KI_DEF

#define KD_DEF

Set these parameters to the values shown in the Tuning Window of the GUI.

Chapter 7 Programming the Microcontroller

If the user wishes to modify the program in the µPD70F3329 device on the microcontroller board, a Flash Programmer is required (not included in the Starter Kit).

Two programmers are available

1. PG-FP5 Full Programmer
2. QB Programmer MiniCube 2 - On Chip Debug / Low Cost Programmer

Figure 7-1 PG-FP5 Programmer

Figure 7-2 QB Programmer (MiniCube2)

The Graphical Interface for either of these programmers can be downloaded from the *NEC Electronics Development tools web site* .

In this guide we have shown only the MiniCube2 (QB Programmer).
The interface for the microcontroller board is the same for both programmers.

Note 1. 1. It can be easier to program the device when the microcontroller board is removed the from the starter kit. Programming the device while the CPU board is connected to the I/O board is possible, although the set up of the programmer is slightly different. The programming operation is still the same. Programming the microcontroller with the board attached to the kit is described in "*section 7.2*" of this user guide.

2. Older programmers such as the PG-FPL or PG-FP4 can still be used with this device. Please refer to the previous version of this user guide for details on how to use this programmer.

7.1 Progamming with the Microcontroller Board disconnected

- Download the following files
  - The MiniCube2 (QB programmer) Flash Programming Graphical Interface
    (from the *NEC Electronics Tool download web site*)
  - The Appropriate IAR Work bench project (with or without the GUI)
    (from the *NEC Electronics Motor Control Starter Kit web site*)
  - Uncompress and Install the QB programmer GUI software (Run the "SETUP" application)
- Save and uncompress the IAR Workbench project
- Check that the switches on the QB programmer are set as shown below
  - a) Switch M1 / M2 is set to "M2"
  - b) Switch 3 - T – 5 is set to "5"
- Then attach the QB programmer to connector J2 on the microcontroller board using the 16-pin cable

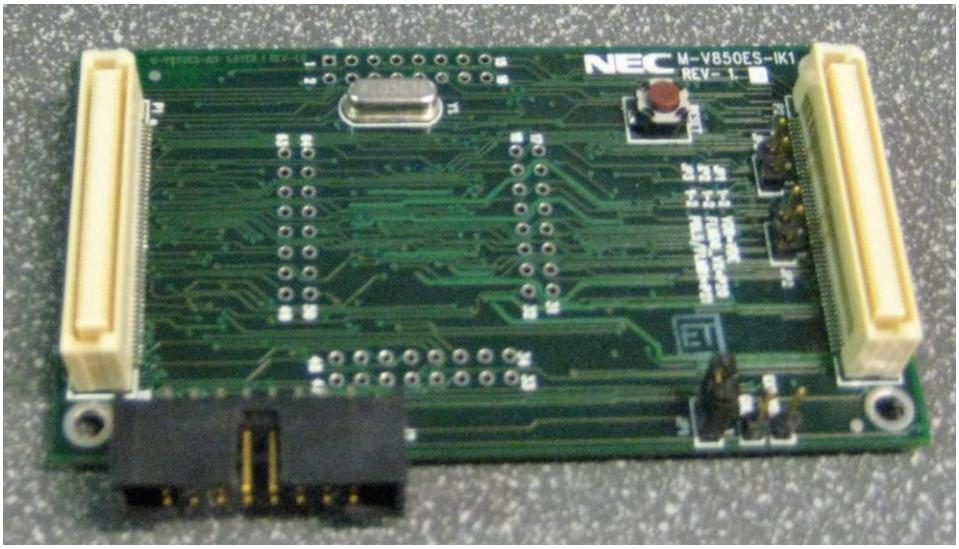

Figure 7-3 Microcontroller Board Flash Programmer Connection

- Copy the 78F0714 Flash programming parameter file (uPD70F3329.prm) into the installation directory for the QB programmer program.

  $installation path$\QBP\....

  (This file is included with the IAR project download. It can be saved in any directory under the “QBP” location)

Once the Microcontroller board is configured and the programmer connected to the PC, open the Flash programming Graphical interface
QBP V2.xx QB-Programmer is run using the windows Start Menu system

(V2.xx will be the revision number of the programmer software)

or

QBP.exe if run via a file manger system

The following Screen should appear. (Note: The text and the device details may differ after the 1st two lines depending on any previous use of the programmer.)

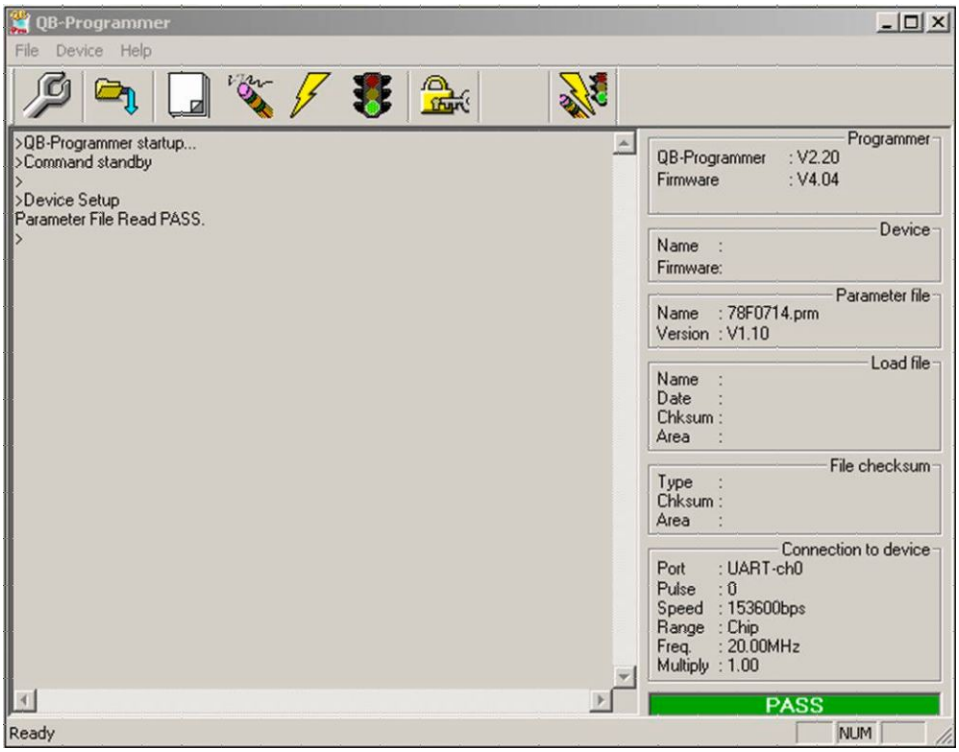

Figure 7-4 MiniCube 2 Programmer GUI (QB Programmer)

Next the device needs to be set up from the menu follow the following sequence:

Device =>Setup…

Or press the "**Spanner**" symbol in the ICON taskbar

The following screen should appear:
(Again the actual details may vary. The exact content is not important at this time.)

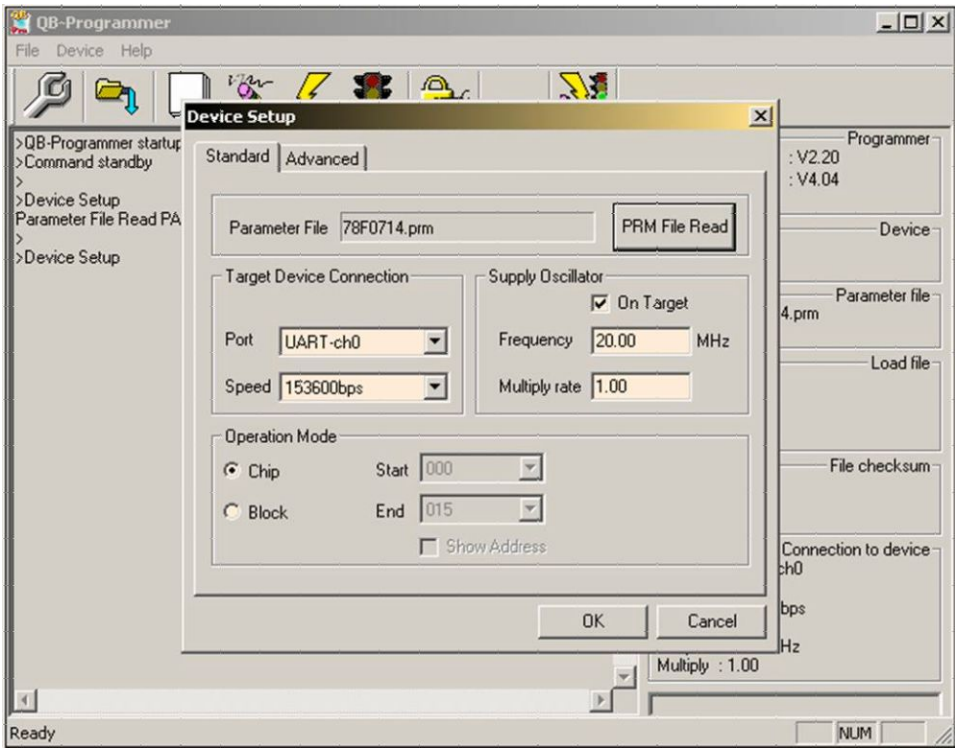

Figure 7-5 Device Setup

Set the details for the COM port and speed as shown above.

Note The device and oscillator are configured by selecting the parameter file as described in the next section below.

To select the correct device parameter file, press the “**PRM File Read**” button and the following screen should appear. If not then find the location of the “PRM-70F3329_Vxxx” directory and the “70F3329.prm” parameter file that was saved when the “ZIP” file was downloaded form the motor control starter kit web site.

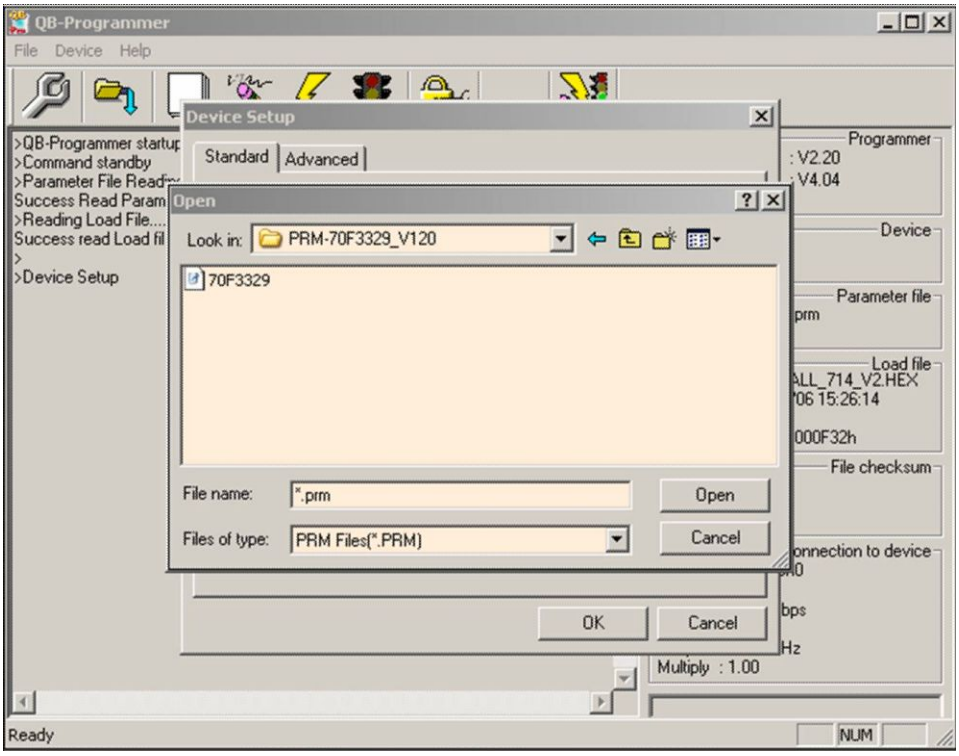

Figure 7-6 Parameter File Read

Select the 70F3329.prm file and press the “**Open**” button

The device details should now be as shown below:

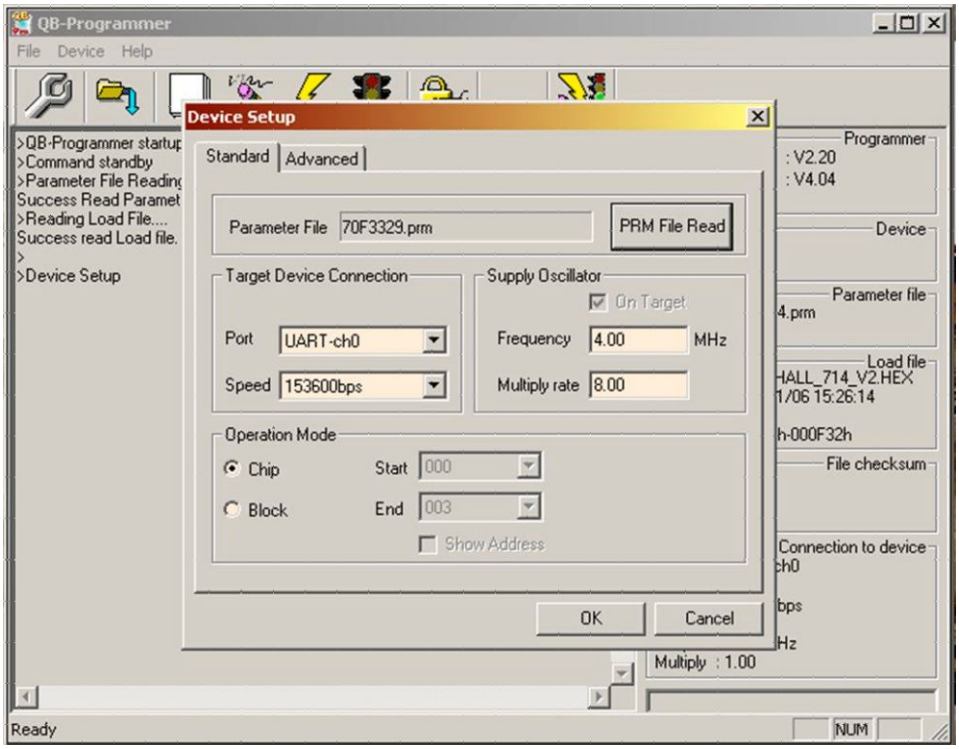

Figure 7-7 Parameter File Read

Then press the “**OK**” button to return to the main menu system

The text in the main screen should read

> Device Setup
Parameter File Read Pass
>

Next select the “HEX” file to be programmed into the microcontroller.

On the File menu press

File => Load

Or press the “Load File” symbol in the ICON taskbar

The following screen should open (Note that the actual details may vary)

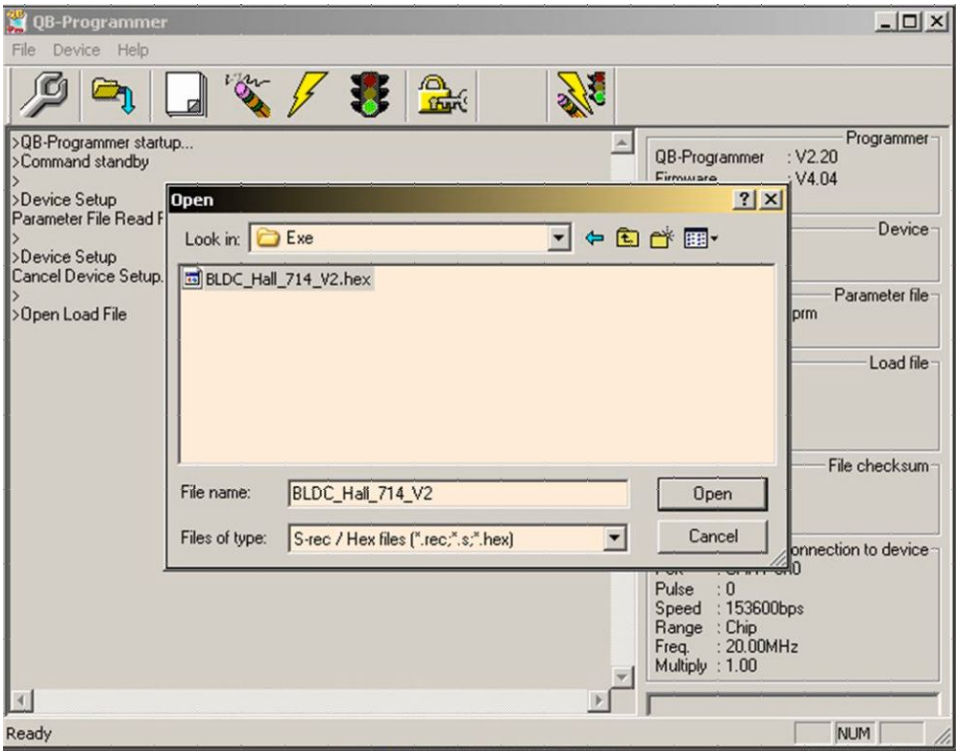

Figure 7-8 Hex File Selection

Locate the “**BLDC_Hall_IK1_GUI.hex**“ file from the saved IAR project

The file can be located as follows

$saved directory$\BLDC_Hall_IK1_GUI\Debug\Exe\....

Select the “BLDC_IK1_GUI.hex” file and press the “**OPEN**” button

The “**Open**” file window will close and the main screen should show the following

> Open Load File
Success read Load file.

The Flash programming setup is now complete.

To program the microcontroller press the “**AUTOPROCEDURE**” button to start the programming sequence.

The following sequence should be seen

>AutoProcedure(Epv)
Blank check Block 000: Not blank, Erase need.
Erasing...
Erase Chip : PASS

```
Program Chip:
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
PASS
Verify Chip:
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
PASS
AutoProcedure(Epv) PASS
>
```

Note The “Verify Chip” operation is enabled by setting the “Read Verify after Program” option in the “Advanced” Tab of the “Device Setup” Window. If this is not enabled the programming sequence will stop after the “PASS” of the “Programming Chip” operation.
The uPD70F3329 device has now been reprogrammed with the example program using the graphical interface. To re program the “stand Alone” again follow the same procedure as above, but download the “BLDC_HALL_IK1” IAR project instead.
The file to be programmed is “BLDC_HALL_IK1.hex”.

7.2 Progamming with the Microcontroller Board connected

The only difference from the procedure defined above, is that the QB programmer switches need to be set as shown below

a) Switch M1 / M2 is set to “M2” (78K0)
b) Switch 3 - T – 5 is set to “T” (Target Power Supply)

Connect the QB programmer as described previously and ensure that once the setup is completed the 15V DC power supply is connected to the starter kit and switched on.

Chapter 8 Using the IAR Embedded Workbench

As already explained the example software for use with the starter kit comprises two versions

- Sensored BLDC control without the Graphical Interface (GUI)
- Sensored BLDC control with the Graphical interface (GUI)

As described above the kit is supplied pre programmed for use in “Stand Alone” mode.
(This is without the GUI)

The complete example project program for the IAR 78K0 Embedded Workbench development tool environment can be downloaded from the motor control web site as detailed in starter kit package (i.e. where this manual was downloaded).

The software is supplied in source format and can be modified as required.

The following sections describe the IAR 78K0 Embedded Workbench development tool environment, how to install it on your computer, and how to rebuild and download executable code to the microcontroller’s flash memory.

Before proceeding with the tools installation, however, refer to all of the documentation for the starter kit, On Chip Debugger tool and the IAR Embedded Workbench.
(Please note that a 16Kbyte code limited version is included with the On Chip Debug unit and can be used to run the example software.)

Please note that a Flash Programmer, On Chip debugging tool or IAR Embedded workbench are not included in this package. These items are available from your local NEC Electronics Distributor or contact your local NEC Electronics sales office.

8.1 Software Installation

1. If a version of the IAR tool is not already installed, then install the IAR embedded Workbench tool as per the instructions provided by IAR.
2. Ensure that if not already that the example software has been downloaded from the NEC starter kit web site and "unzipped" into an suitable location.

The example software can operate on any revision of either the IAR Kick Start or IAR Full versions. However it may be necessary to define your own project and workspace. This is described later in this chapter.

8.2 MC-LVKIT-IK1 Setup for On-chip Debugging

To set up the MC-LVKIT-IK1 starter kit for debugging, follow the steps below.

1. Disconnect the M-70F3329 micro board by separating it from the MC-IO board.
2. Ensure that links SB9 and SB11 (UART RxD and TxD) are connected. (This should be the default setting.)

3. Ensure jumpers JP2 and JP3 are removed.
(Unless the alternate use of the FLMD0 and FLMD1 pins are used.)

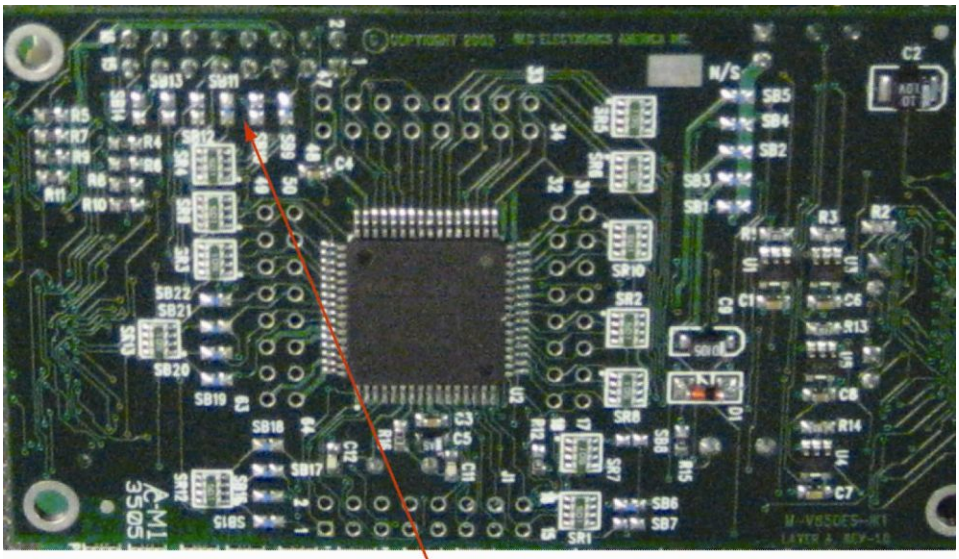

Figure 8-1 Microcontroller Board OCD Configuration

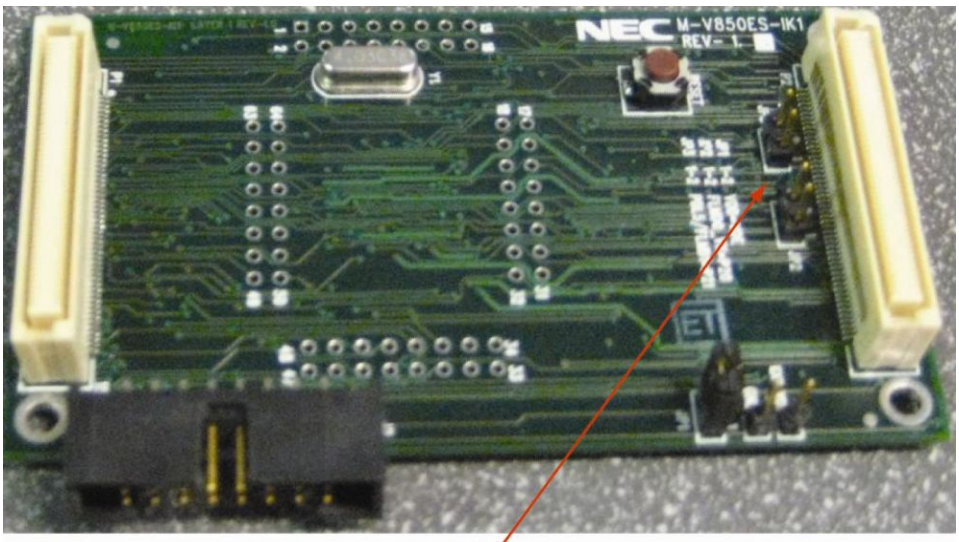

Figure 8-2 Microcontroller Board OCD Configuration

4. Reconnect the MC-CPU-70F3329 board to MC-IO board
5. Attach the 16pin on-chip debugging emulator's target cable to connector J2 on the MC-CPU-70F3329 and connect the USB cable between your computer and the MiniCube2 as shown bellow.
(Note this is the same connector used for programming).

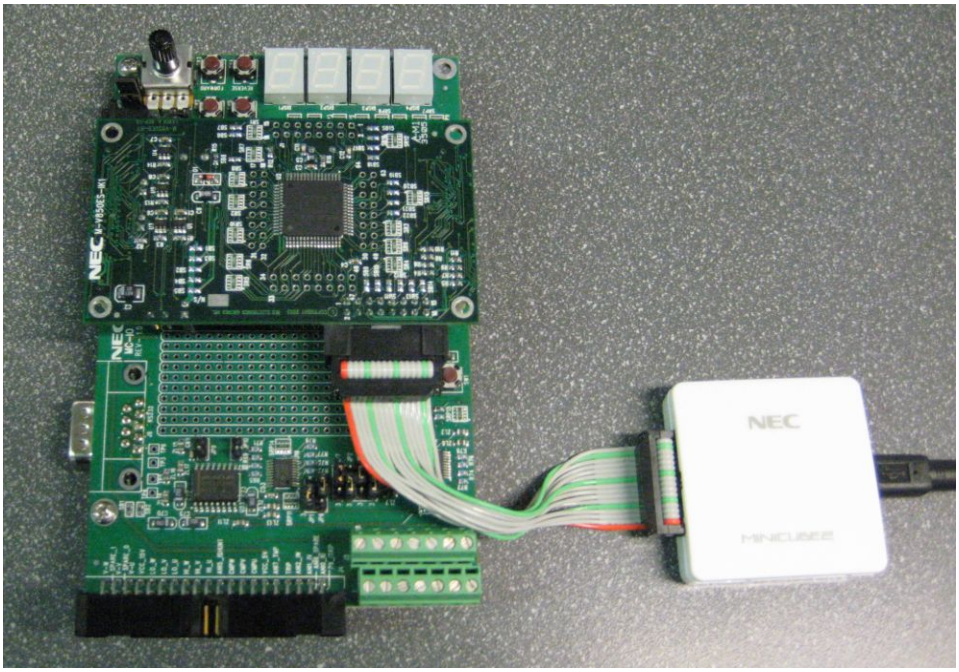

Figure 8-3 OCD Debugger Connection

The microcontroller on the 70F3329 CPU board operates internally at 32 MHz, which requires an external clock of 4 MHz. This is already supplied directly from the crystal on the board. The MiniCube2 needs to be set with the same frequency so that the UART baud rate communication is the same for both the device and MiniCube2. The clock can be set internally to the MiniCube2 without any external oscillator or crystal connected to the OCD unit.

Please refer to the MiniCube2 On-Chip Debugger user's manual for information about how to configure the application for V850 devices (uPD70F3329).

Please note that to use the MiniCube2 for on chip debugging a background debug monitor is used which requires some additional resources to be used in the V850ES/IK1. If modifying the application code the user should ensure that these are available if the on chip debug is used.

Two additional files are required to reserve the interrupt functions and memory area. These are

1. monitor_dbg0.s85 – This assembler file reserves the Interrupt vector for the background monitor program
2. monitor_csi0.s85 – This assembler file reserves the interrupt vector for CSI0 used for communicating between the miniCub2 and the target device.

The background monitor is loaded automatically by the IAR C-Spy debugger and is located at the end of the Flash memory area. It is necessary to ensure sufficient space is available for the monitor. (Refer to the MminiCube2 user manual for explanation of the resources used by the monitor program.)

The above assembler files are not required to operate the V850ES/IK1 starter kit in stand alone mode and have been removed from the build and project. Please follow the instructions in section 8 (creating a new project), to add these file to the project. It is not necessary to create a new project to add these file if the downloaded sample IAR software is compatible to your IAR workbench installation. The files can just be added and the project rebuilt before using the MiniCube2 OCD unit.

8.3 IAR Embedded Workbench

1. Open the IAR workbench. The following Screen should be opened: (Note the exact display may vary depending on if this is a new installation)

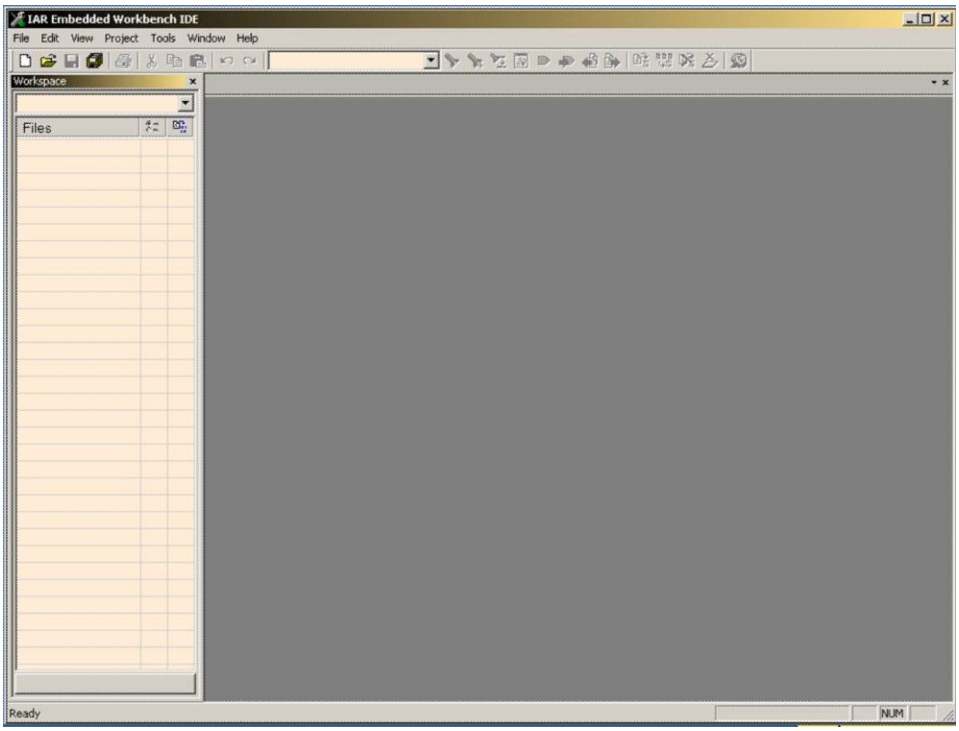

Figure 8-4 IAR Workbench Opening Screen

2. Next open the IAR Workspace by following the sequence and locating and then selecting the appropriate workspace file as shown below:

   File -> Open -> Workspace -> BLDC LVSK IK1

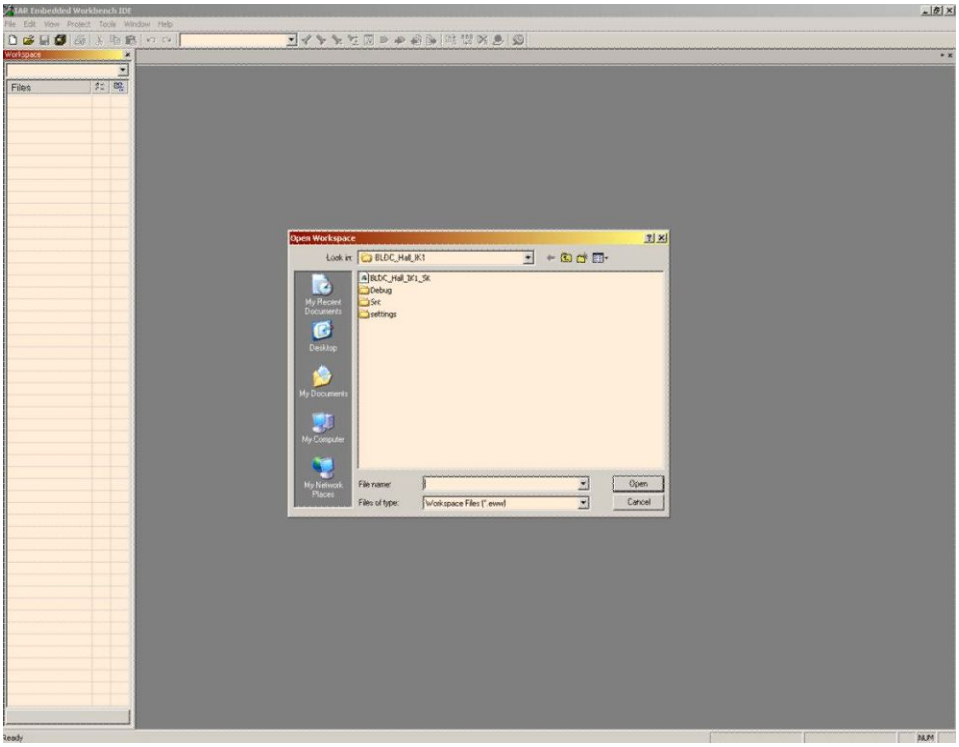

Figure 8-5 IAR Workbench Project Selection

Once the Workspace open the display should look something close to that as shown in *Figure 8-5*.

This shows the workspace where the project is located and has opened either the BLDC without GUI or BLDC with GUI project. The display shows the following project files

- Left Hand side window – Project File (Source. Header, Map etc)
- Bottom Build Debug messages when the project is re built or the debugger is active
- The main centre display shows any open files in a tabbed form. The file can be viewed by selecting the relevant Tab in the wind

Any of the files shown can be opened by double clicking on the file in the "Project" (left hand side) window. Debugging windows are described later.

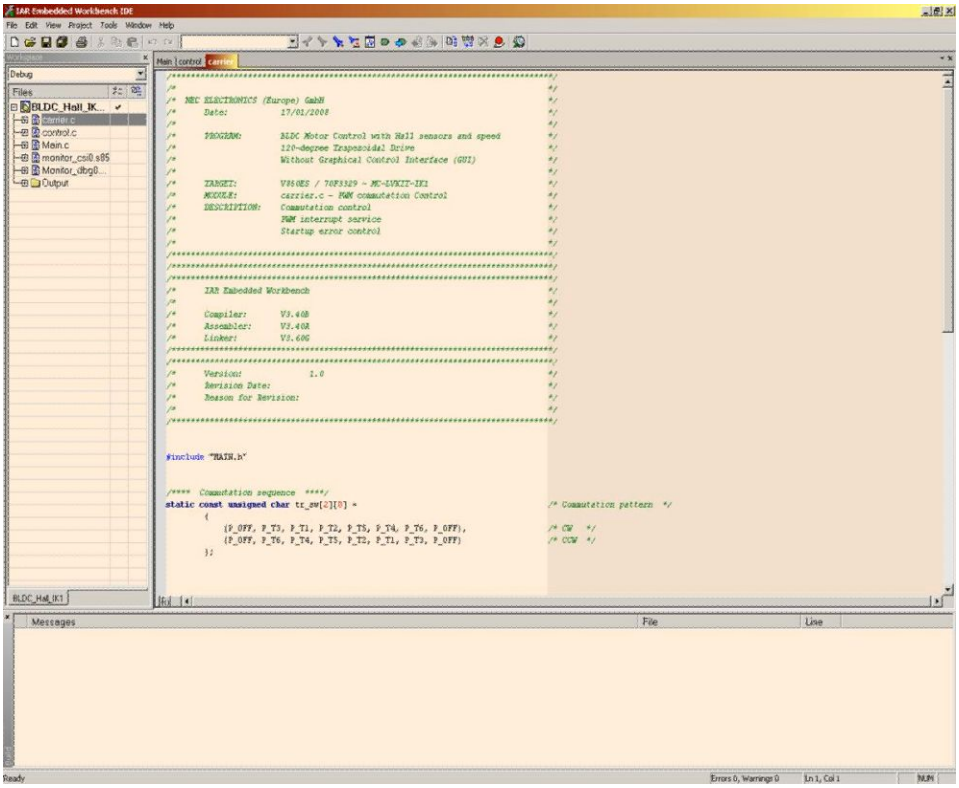

Figure 8-6 Workspace & Project open screen

3. The build options for the project can then be set or changed using the following menus. The build options are entered as shown in "*Figure 8-7*" below, then ensure all the options are as shown in "*Figure 8-8*" to "*Figure 8-11*" below.

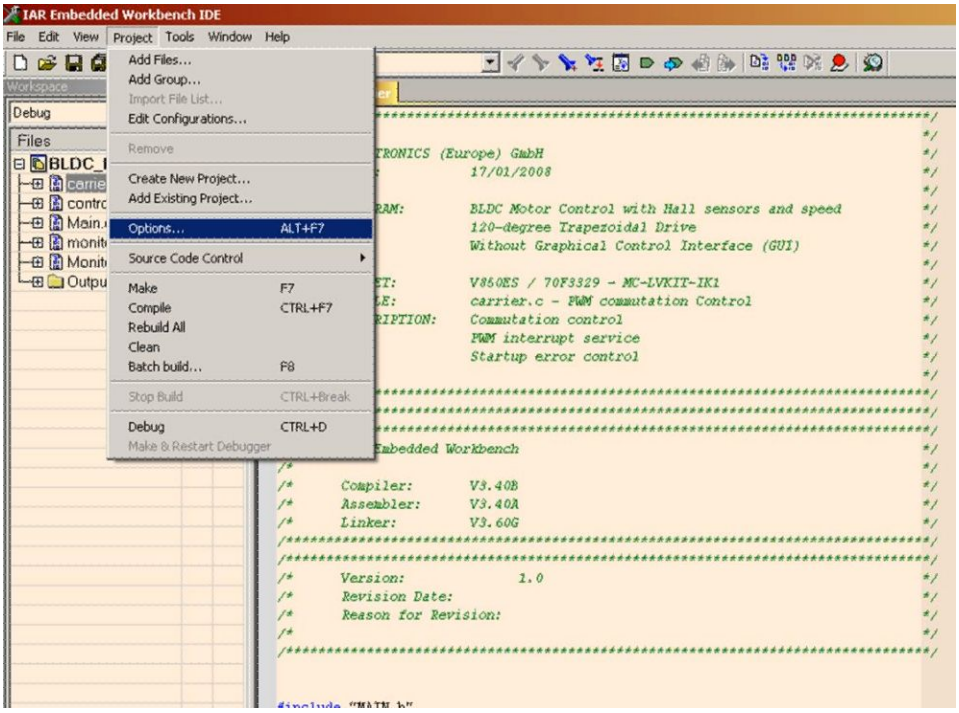

Figure 8-7 Project build options

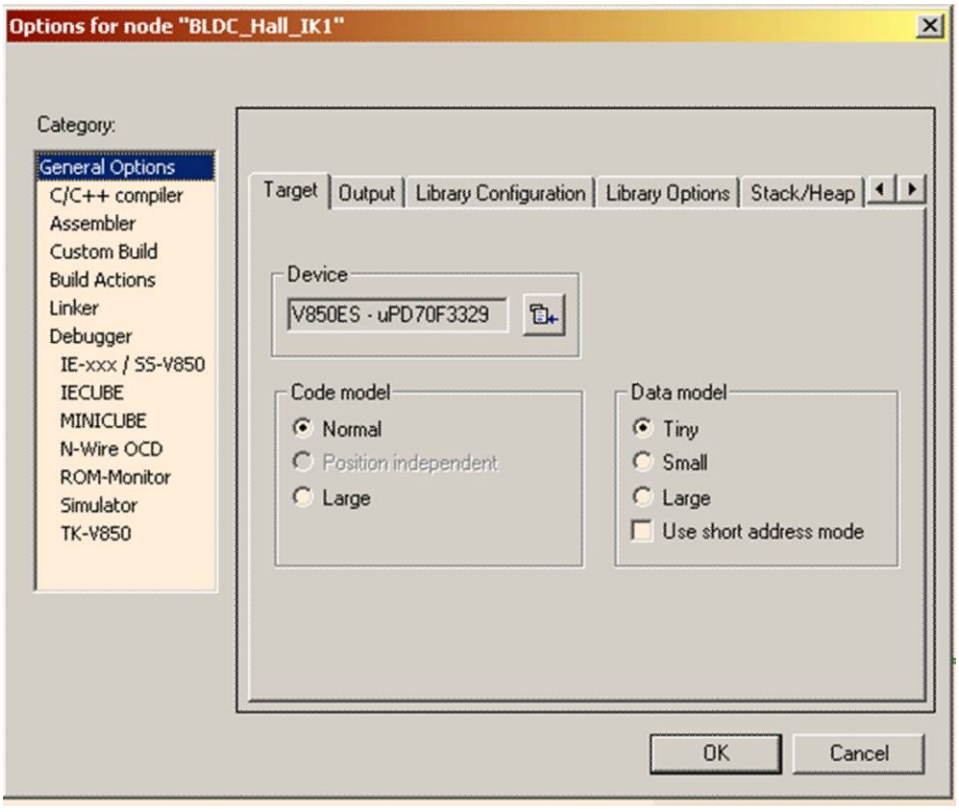

Figure 8-8 General Options – Setting the Target Device

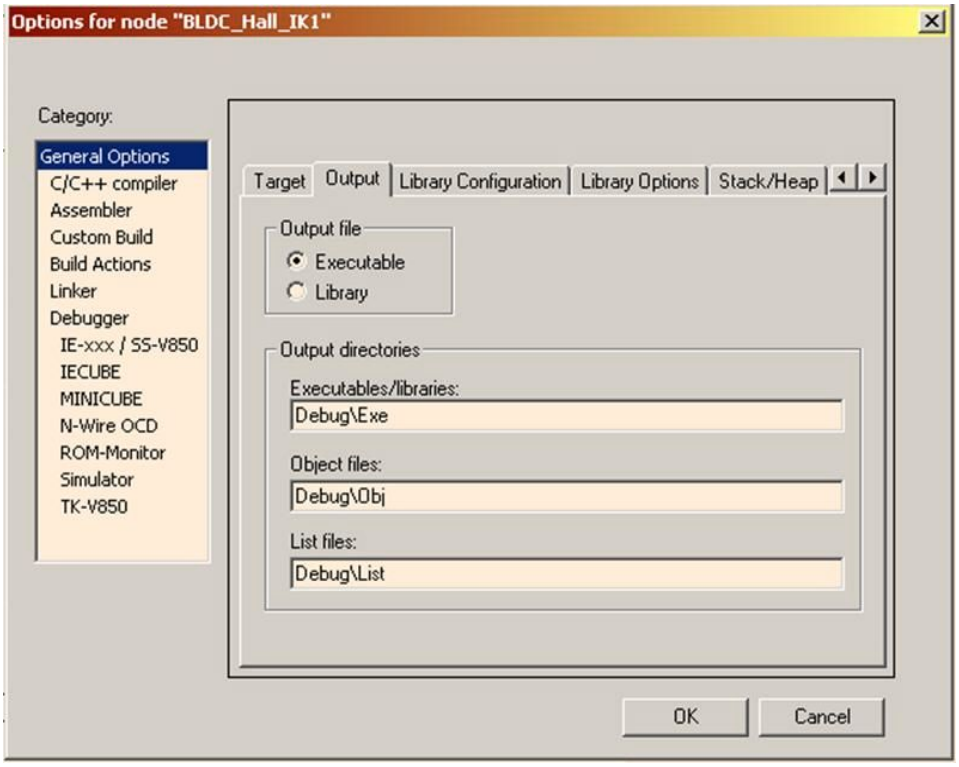

Figure 8-9 General Options – Setting the Output Locations

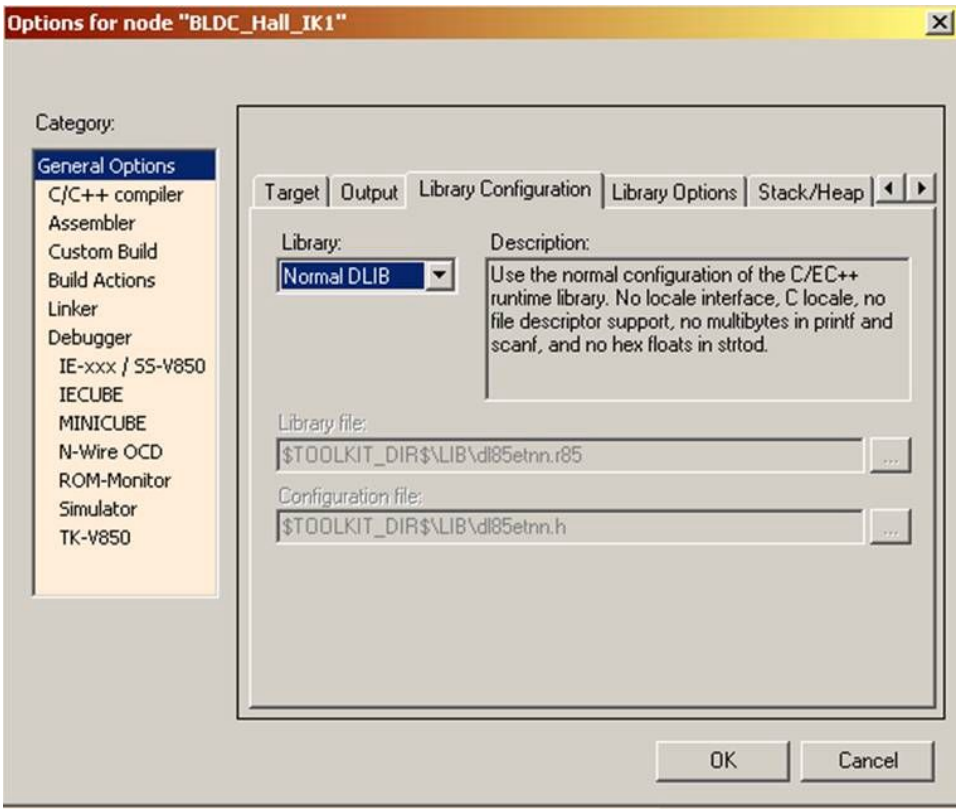

Figure 8-10 General Options – Selecting the C-Library

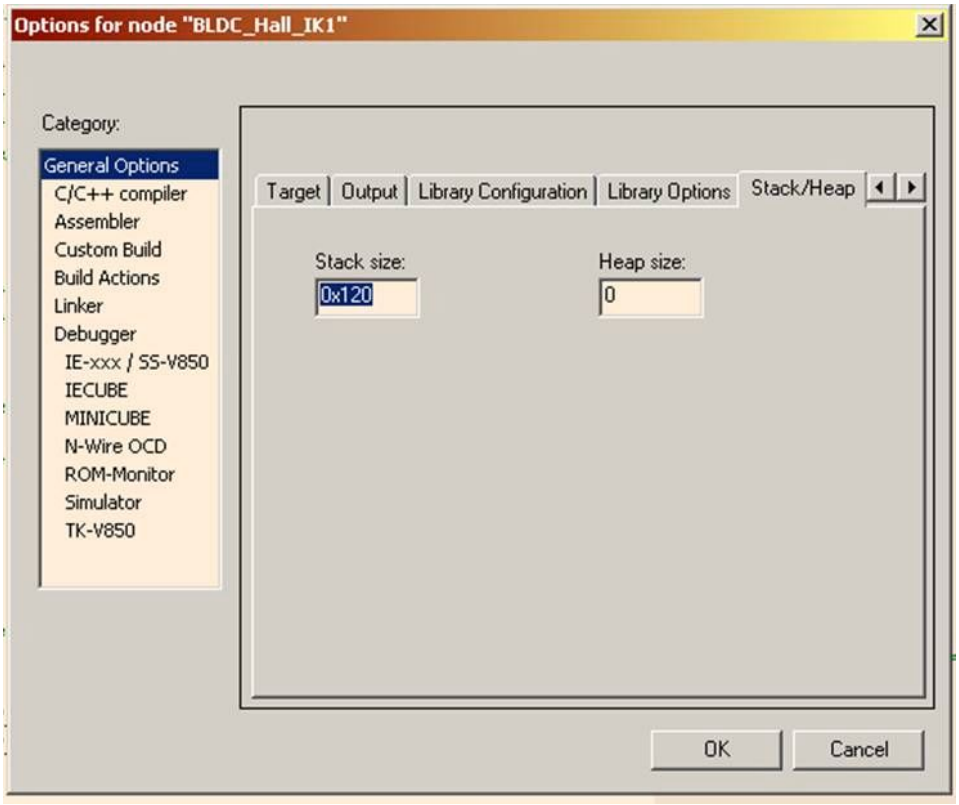

Figure 8-11 General Options – Setting the Stack and Heap

8.4 Compiler Options

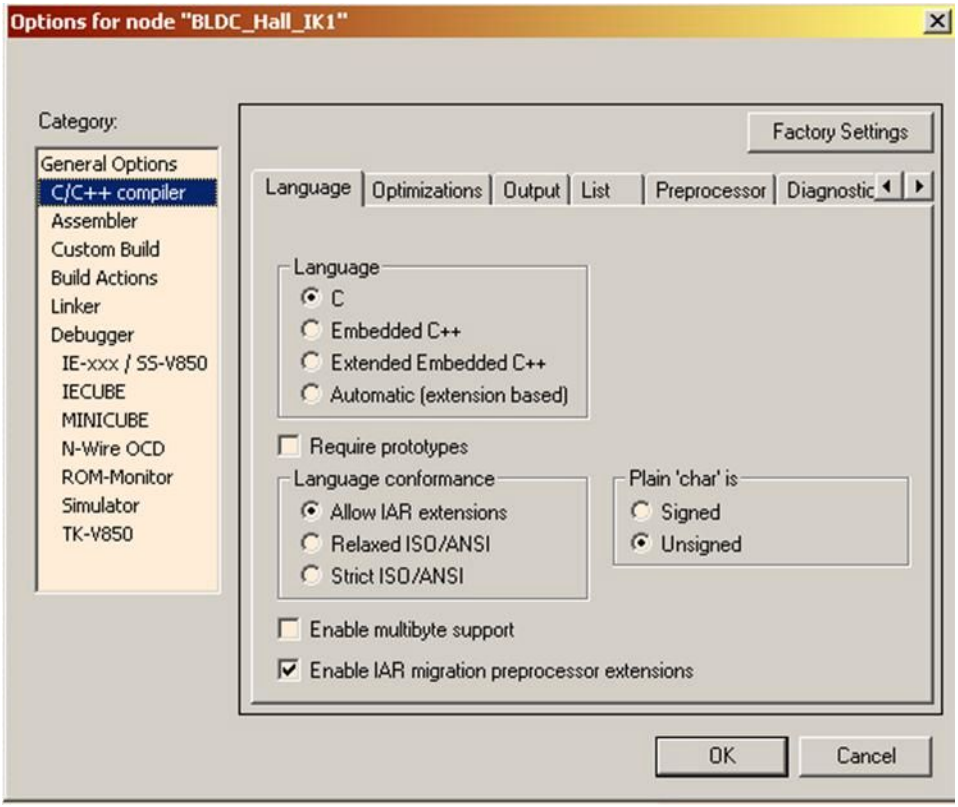

Figure 8-12 Compiler Options – Language Settings

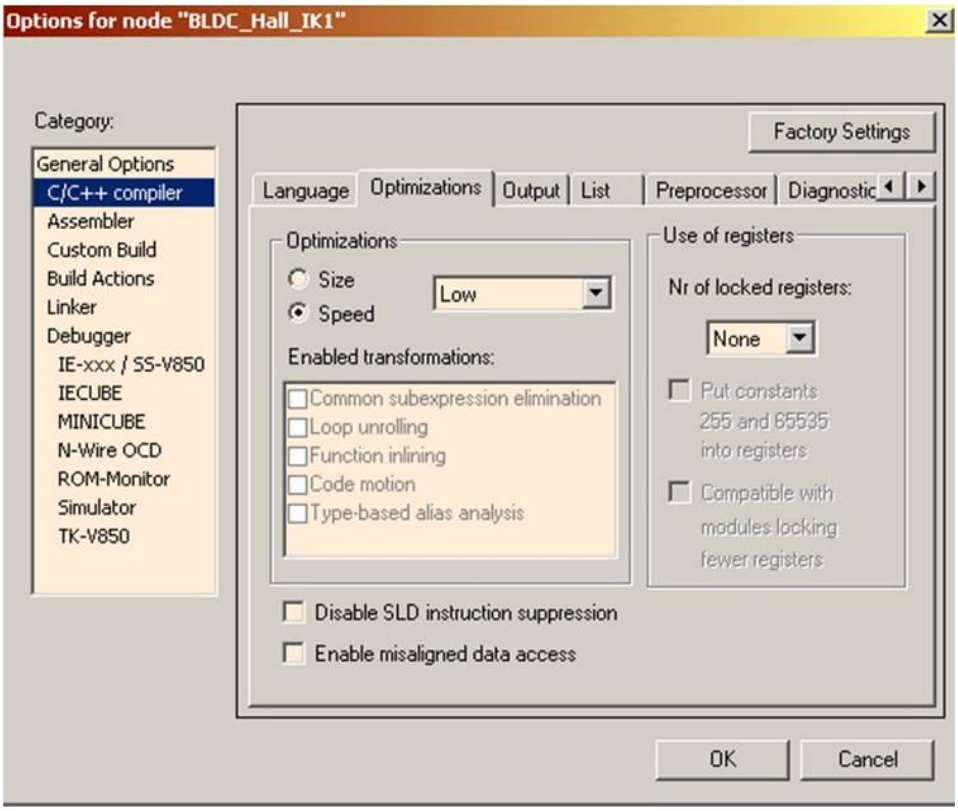

Figure 8-13 Compiler Options – Optimisation

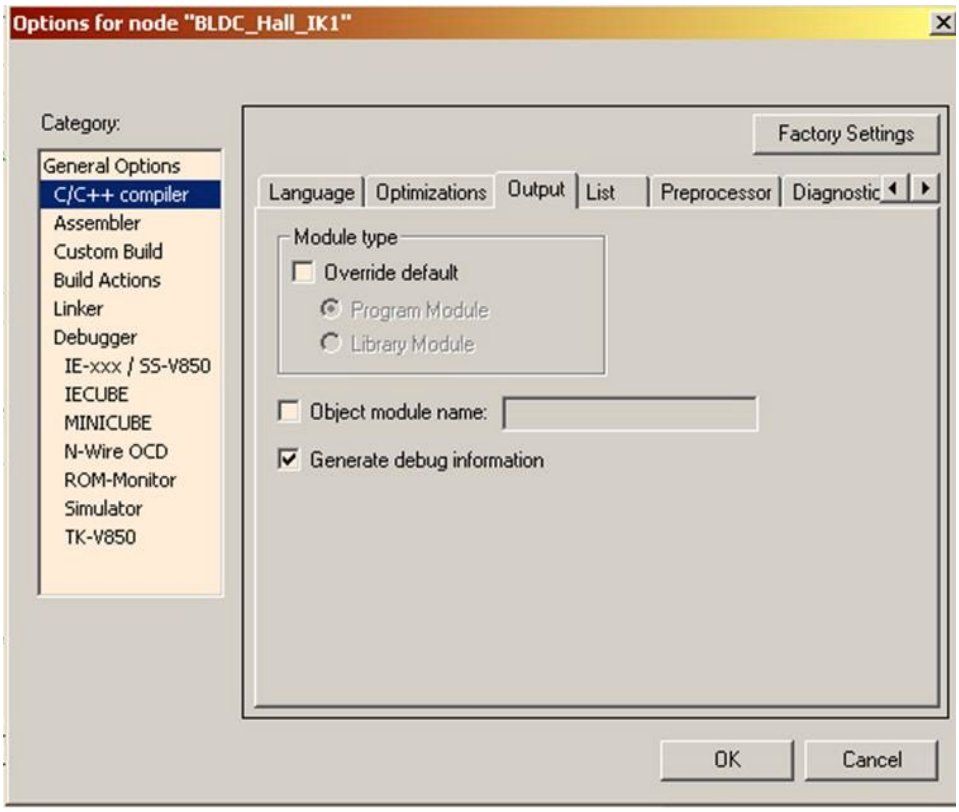

Figure 8-14 Compiler Options – Output Set for Debug

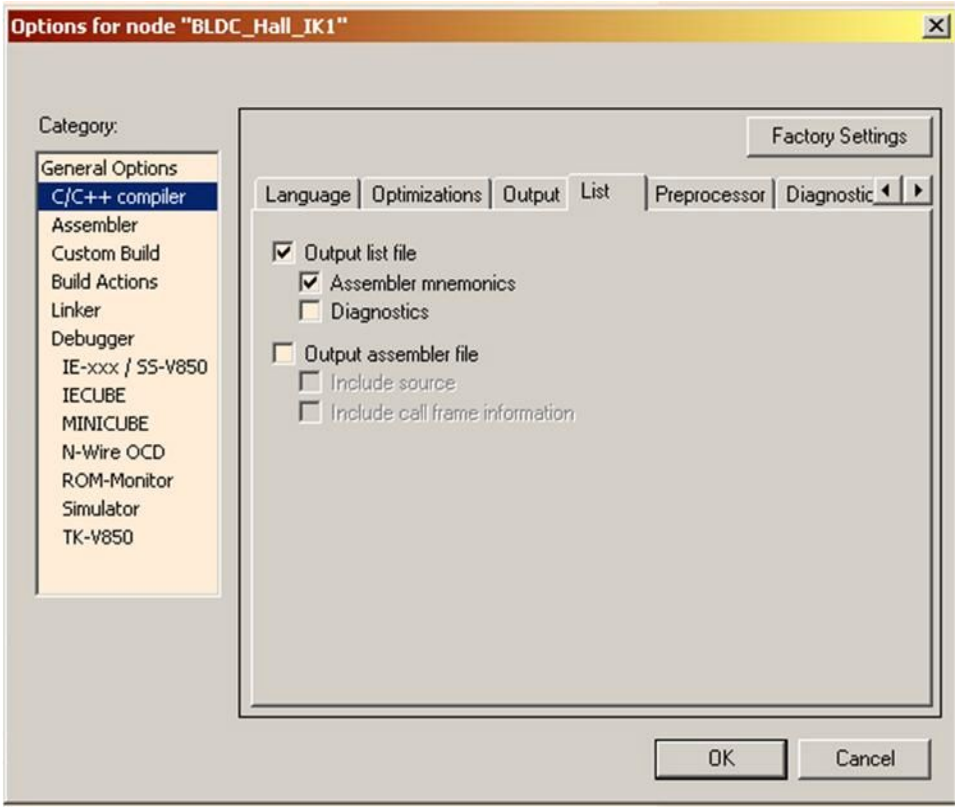

Figure 8-15 Compiler Options – Compiler Listings

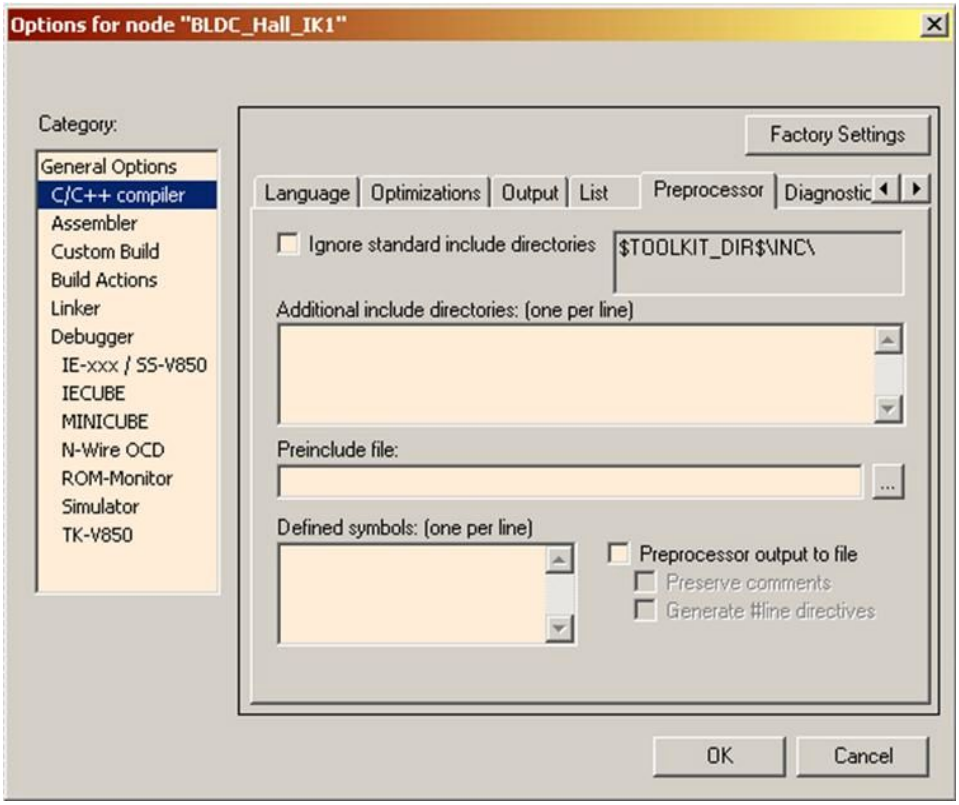

Figure 8-16 Compiler Options – Pre Processor Settings

Note All other Compiler Options settings can be remain as the default settings.

8.5 Assembler Options

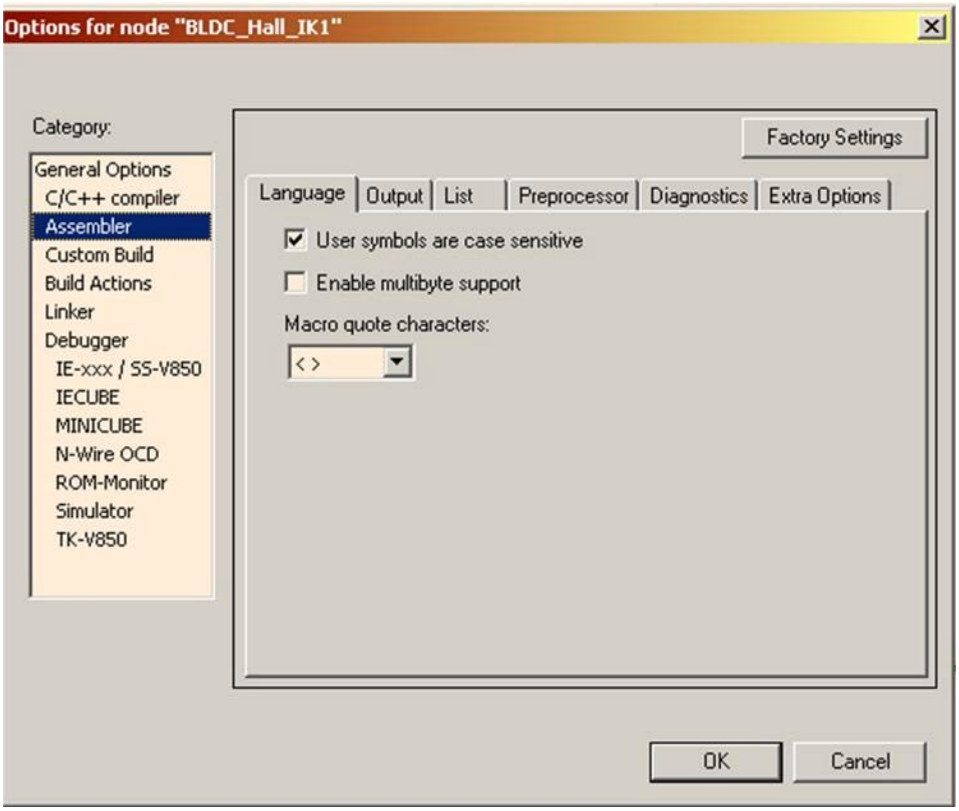

Figure 8-17 Assembler Options – Language Settings

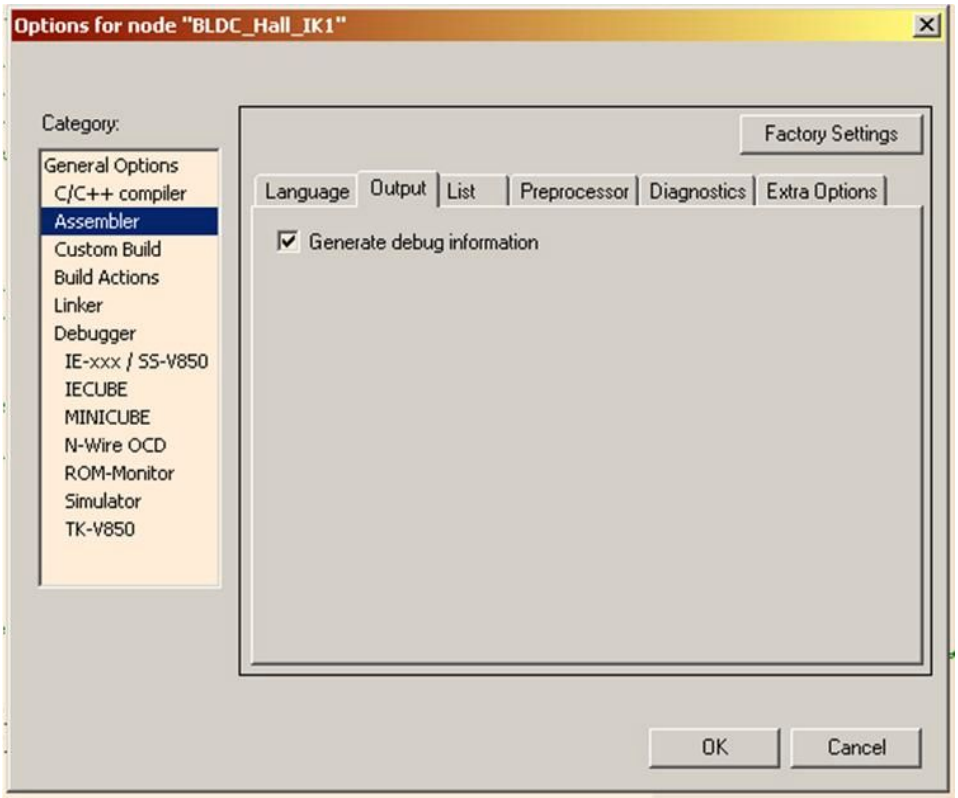

Figure 8-18 Assembler Options – Output set for Debug

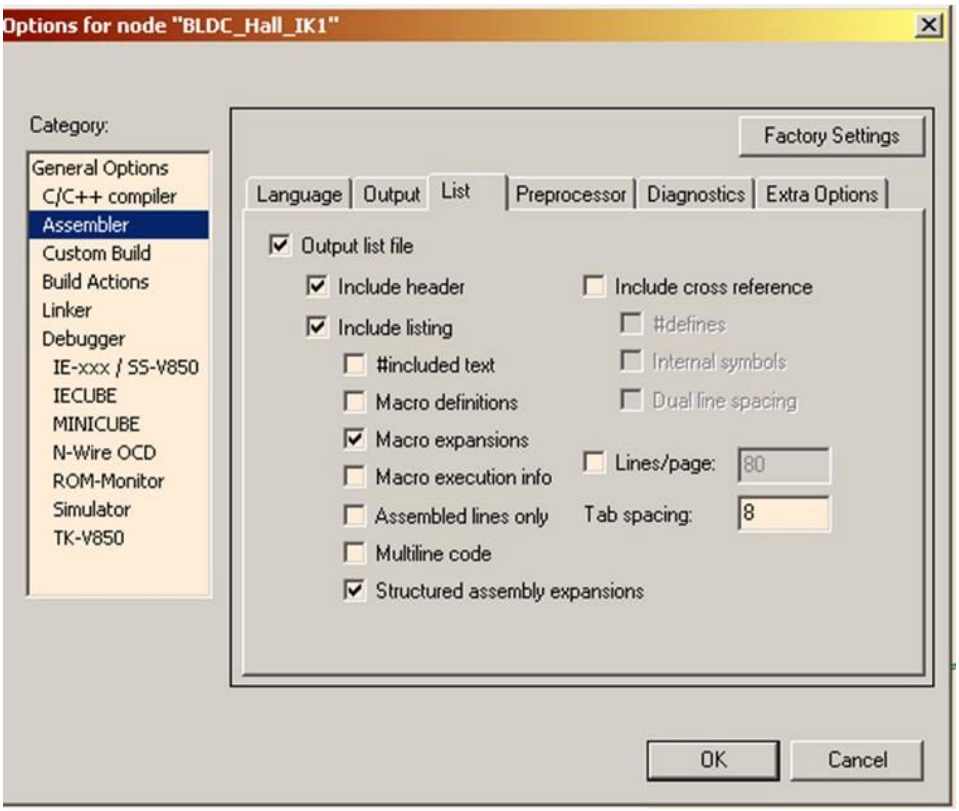

Figure 8-19 Assembler Options – Listings

Note All other Assembler Options can be left as the default setting.

8.6 Linker Options

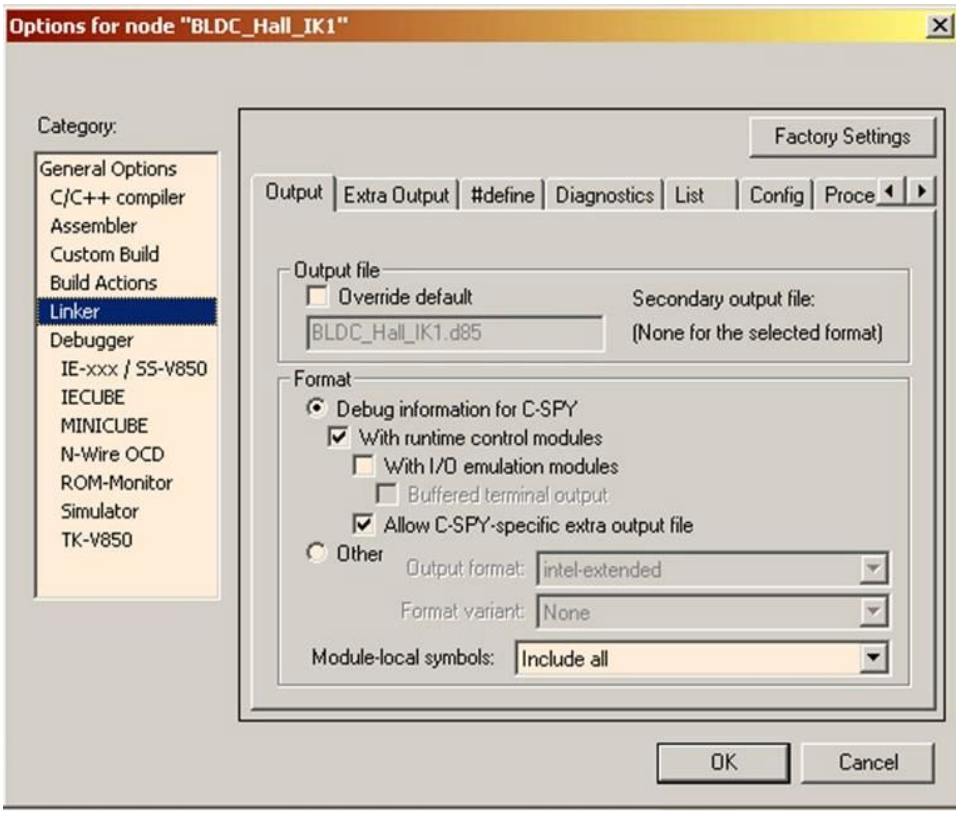

Figure 8-20 Linker Options – Primary File Output

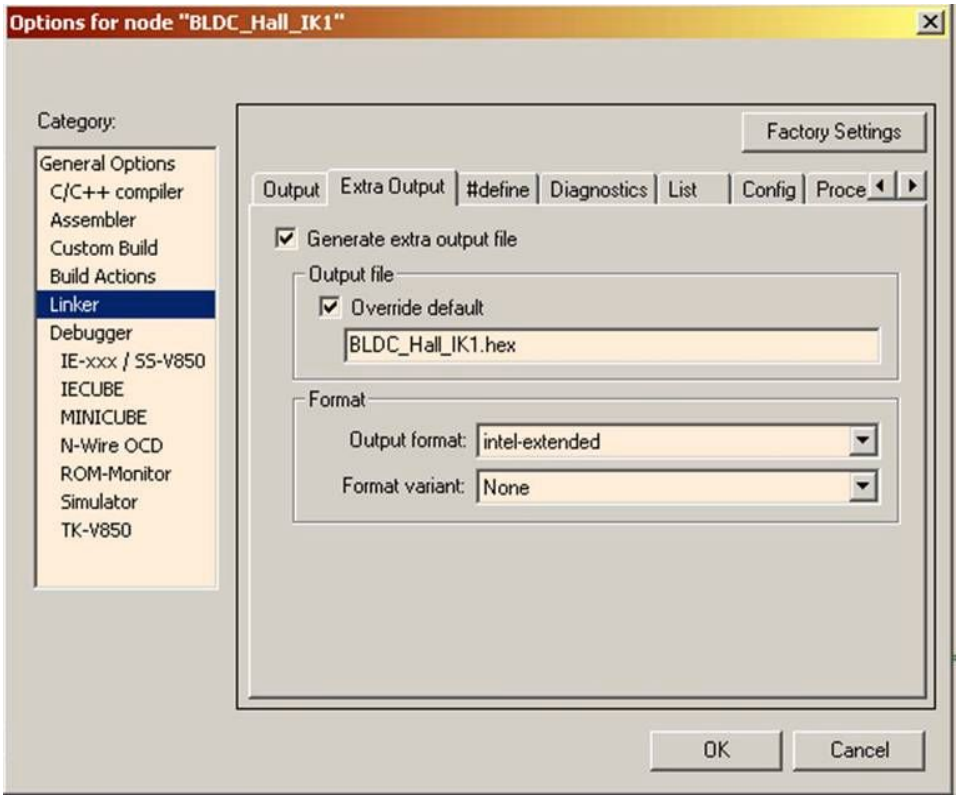

Figure 8-21 Linker Options – Secondary File Output

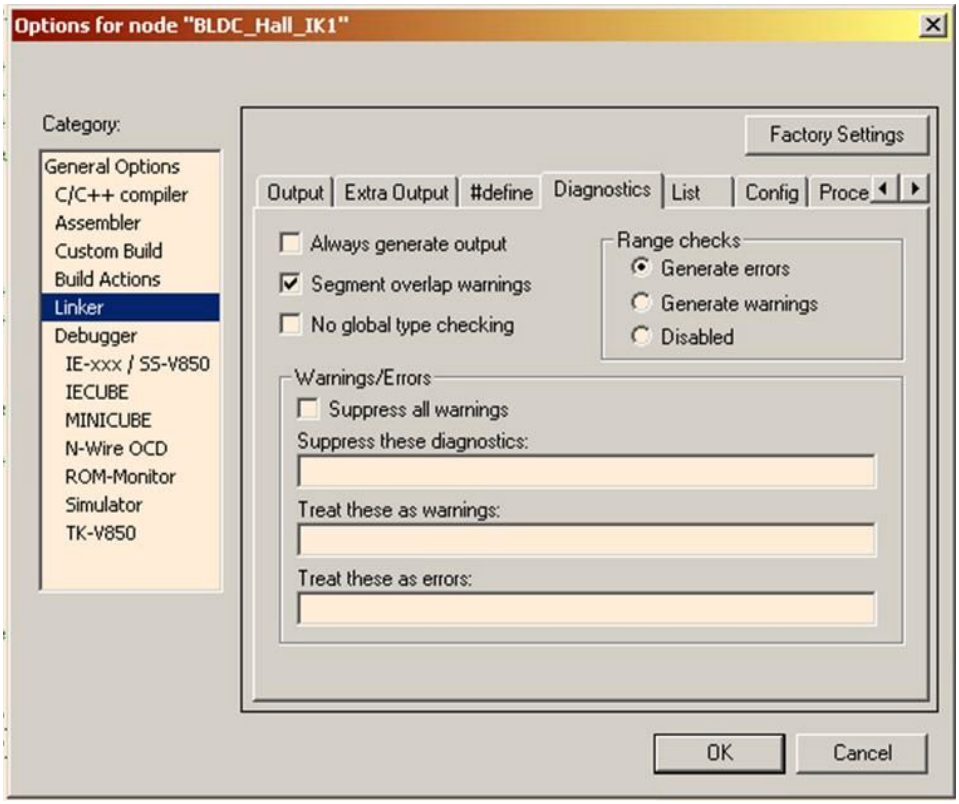

Figure 8-22 Linker Options – Diagnostic settings

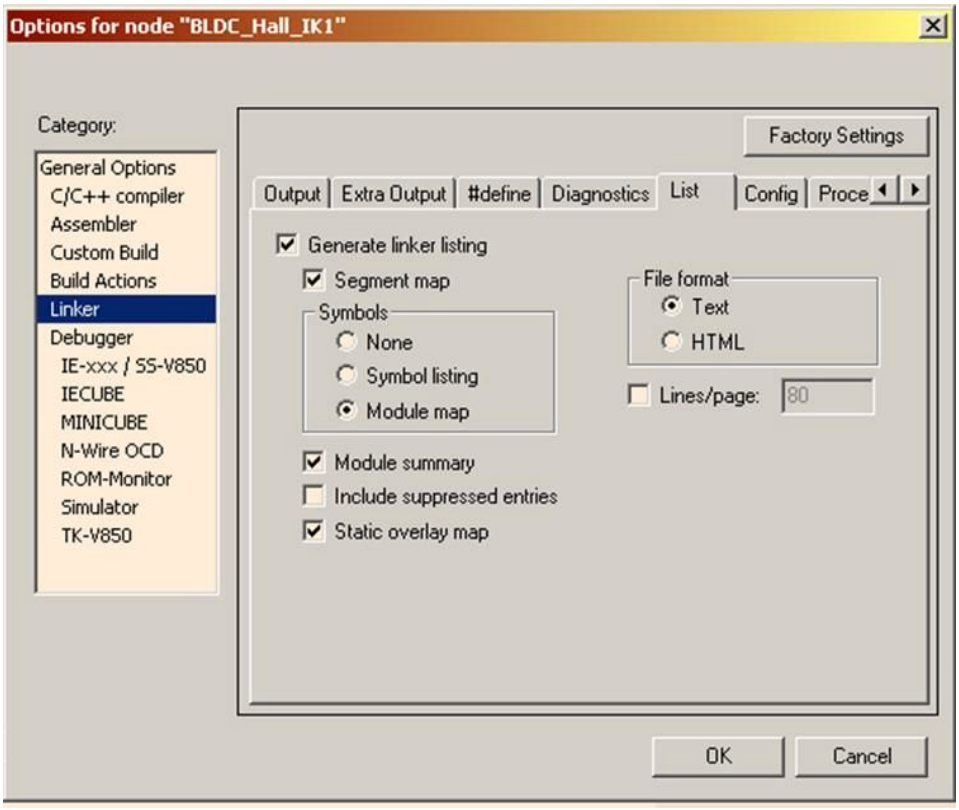

Figure 8-23 Linker Options – Generate MAP File Output

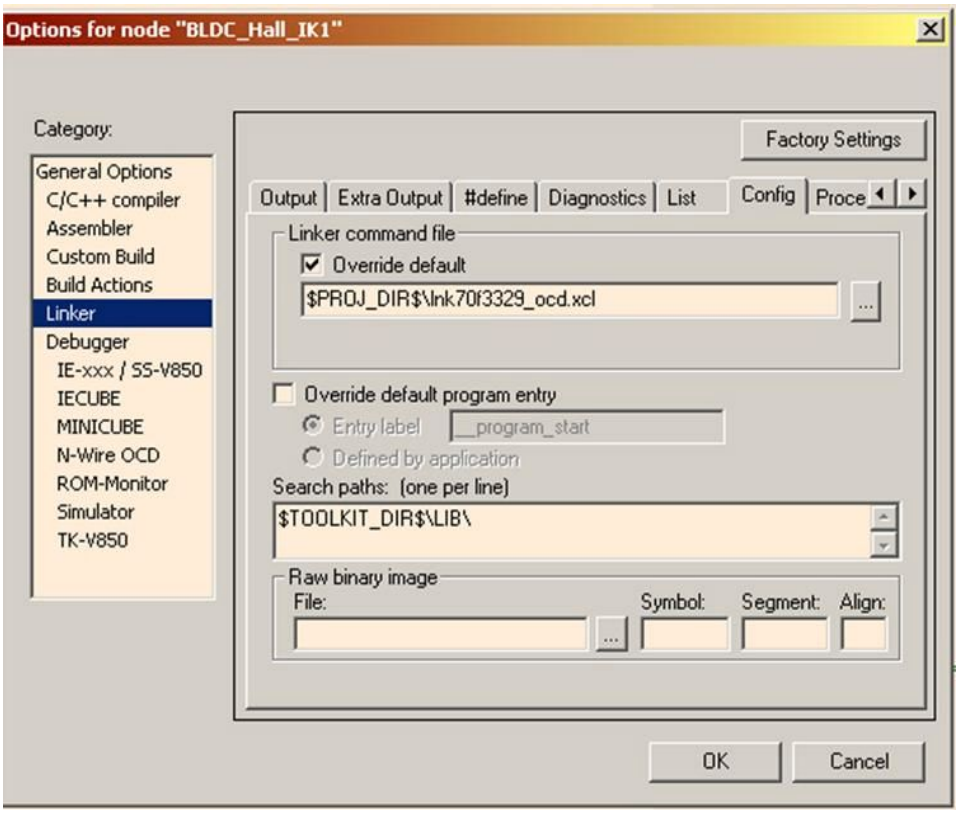

Figure 8-24 Linker Options – Linker Command File Selection

Note The Linker Control File must match the device selected.
The remainder of the Linker setup can be left as the default setting.

8.7 Integrated Debugger Selection

Note The Extra Option section can be ignored.

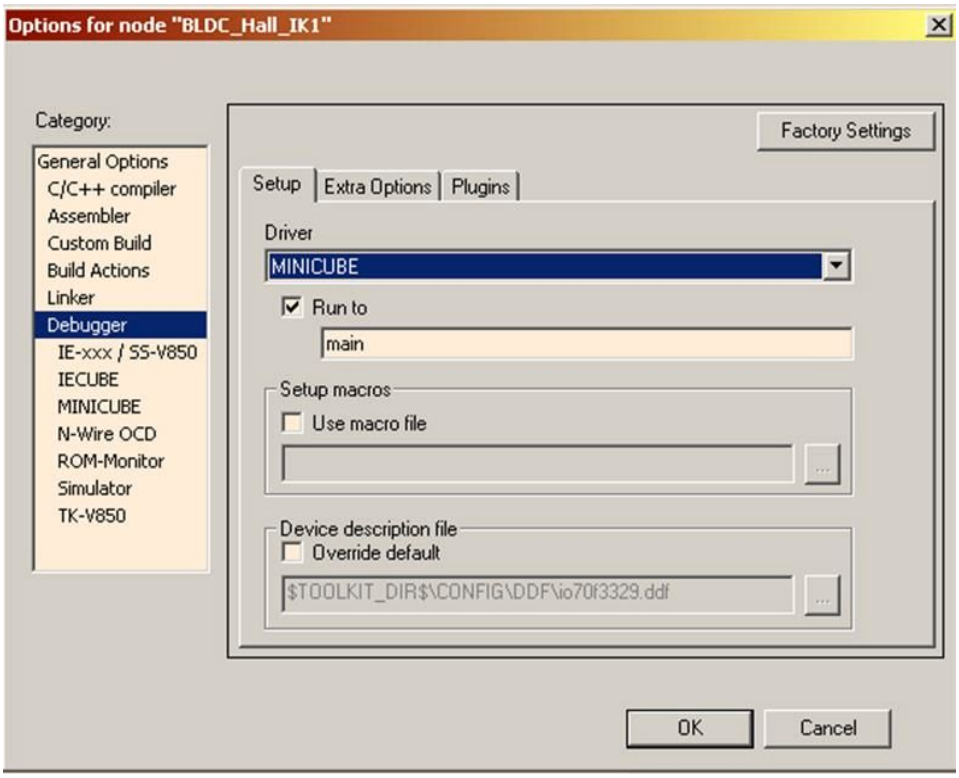

Figure 8-25 Integrated Debugger Selection

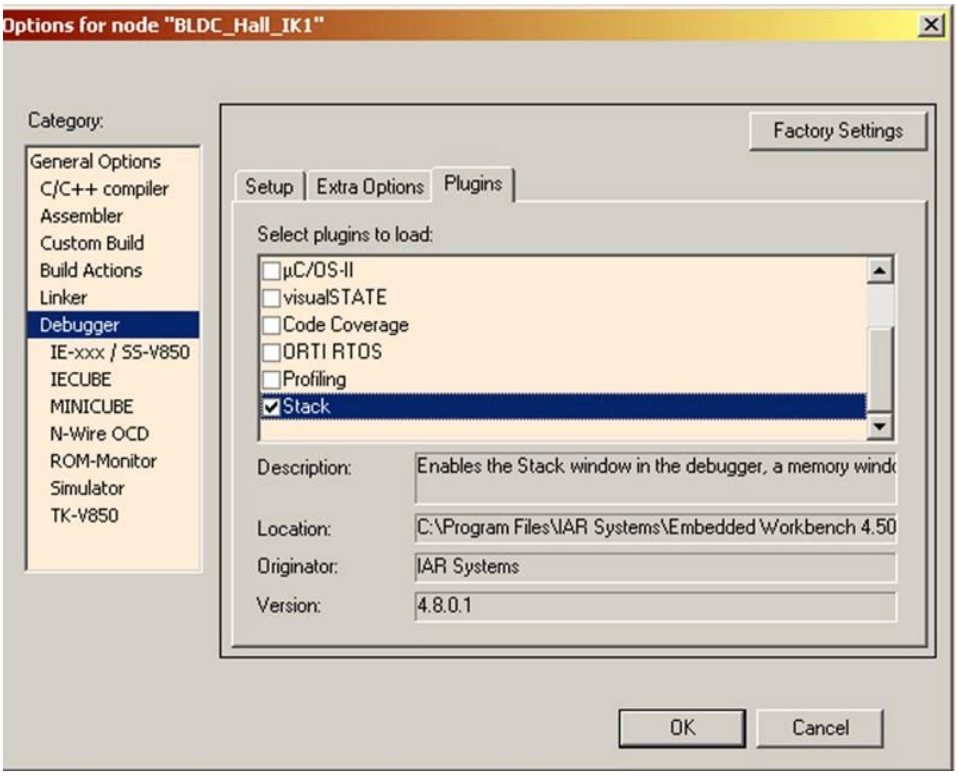

Figure 8-26 Integrated Debugger - Plug-in Selection

Note All other plug-in options should not be selected.

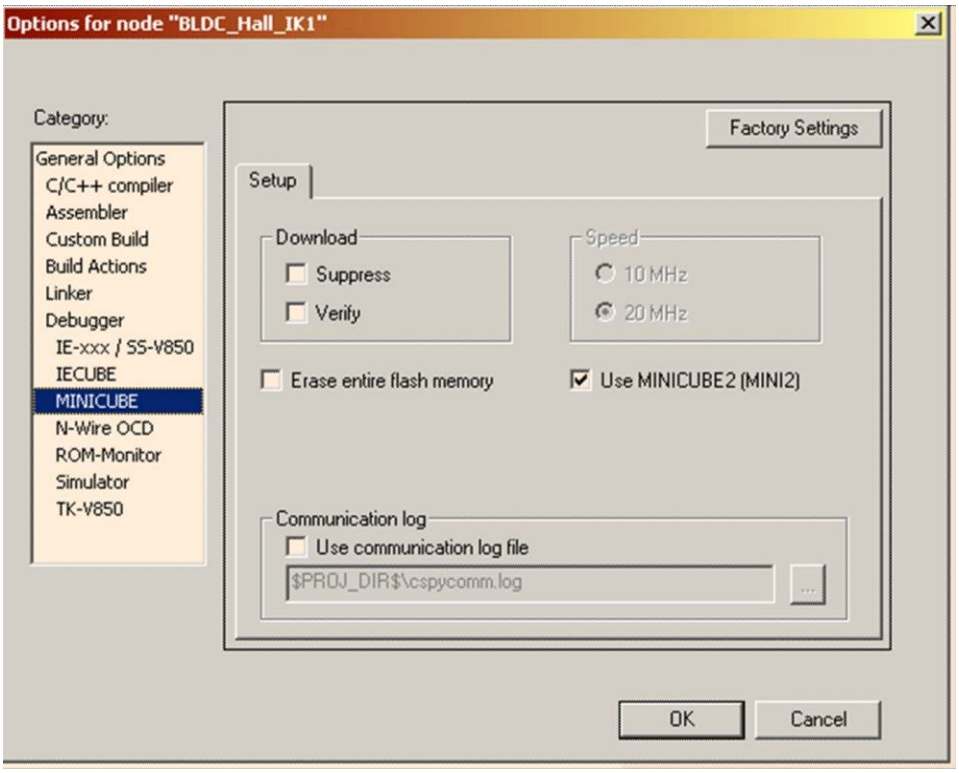

Figure 8-27 Integrated Debugger - MiniCube2 Selection

8.8 Workspace and Project Setup if the example is not compatible with the installed IAR workbench

- Set a new workspace
  File -> New -> Workspace

- Create a new Project
  Project -> Create New Project -> Select "Empty Project" -> "OK"

  Enter a project name and set the location for the project
  (This can be the same location as the downloaded example software or a new location)

- Add the Source files to the project
  - C Source Files
    Project -> Add Files

    Locate and select all the C source files
    - **Main.c**
    - **Control.c**
    - **Carrier.c**

      Press "OPEN"

  - Repeat the operation for the Assembler files
    Project -> Add Files

    When the Selection window opens select
    Files of Type -> Assembler Files

    Select the assembler files only if On Chip Debugging is required.
    (If OCD is not needed at this time then these files can be omitted.)
    - **monitor_csi0.s85**
    - **Monitor_dbg0.s85**
    -

      Press "OPEN"

All these files should now appear in the project window (left hand side of the IDE), as shown previously.

8.9 Build / Rebuild the Project

To build the project press the “make” icon in the task bar as shown below:

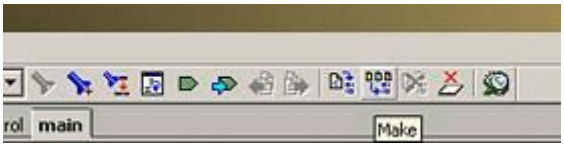

Figure 8-28 Make Button

The build results and any errors or warnings will be displayed in the Messages window at the bottom of the IAR Workbench window. These should be corrected before moving on to the Debugging section.

8.10 Debugging

Once the project has been built without errors the user can now start the debugging session.

This is done by pressing the "Debug" icon in the task bar.

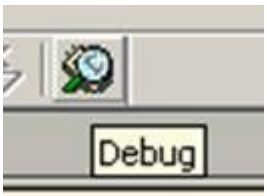

Figure 8-29 Start the Debug Session

The debugger will connect to the OCD unit and download the code to the Flash memory on the microcontroller board.

Once downloaded the debugging window will open as shown below in *Figure 8-0* .

Note The IAR embedded Workbench provides an integrated debugger, so the debugging window opens as part of the IDE.

If the debugger is run for the first time in a new project the following set up window will open. This is to set the basic function of the debugging hardware (i.e. Mini Cube or IECube etc.).

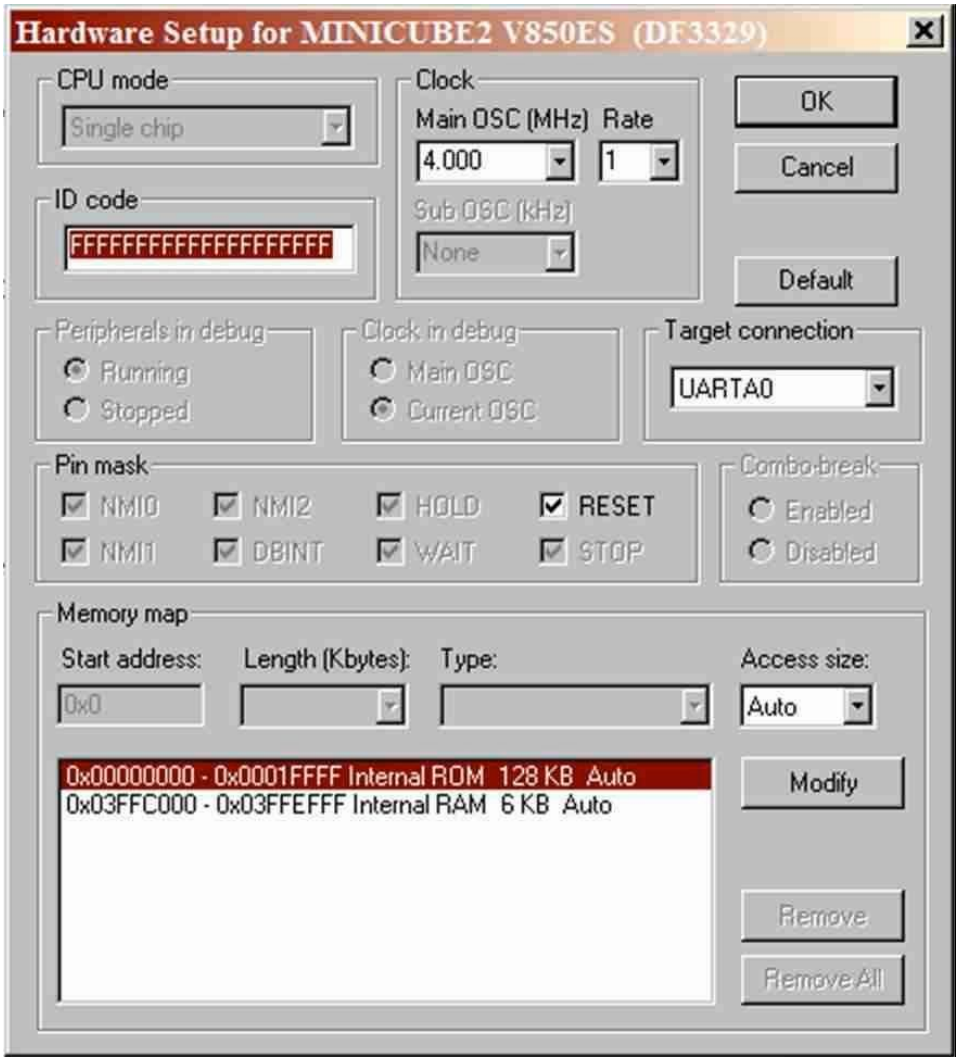

Figure 8-30 Debugging – Initial Hardware Setup

Ensure that the settings are as defined above.

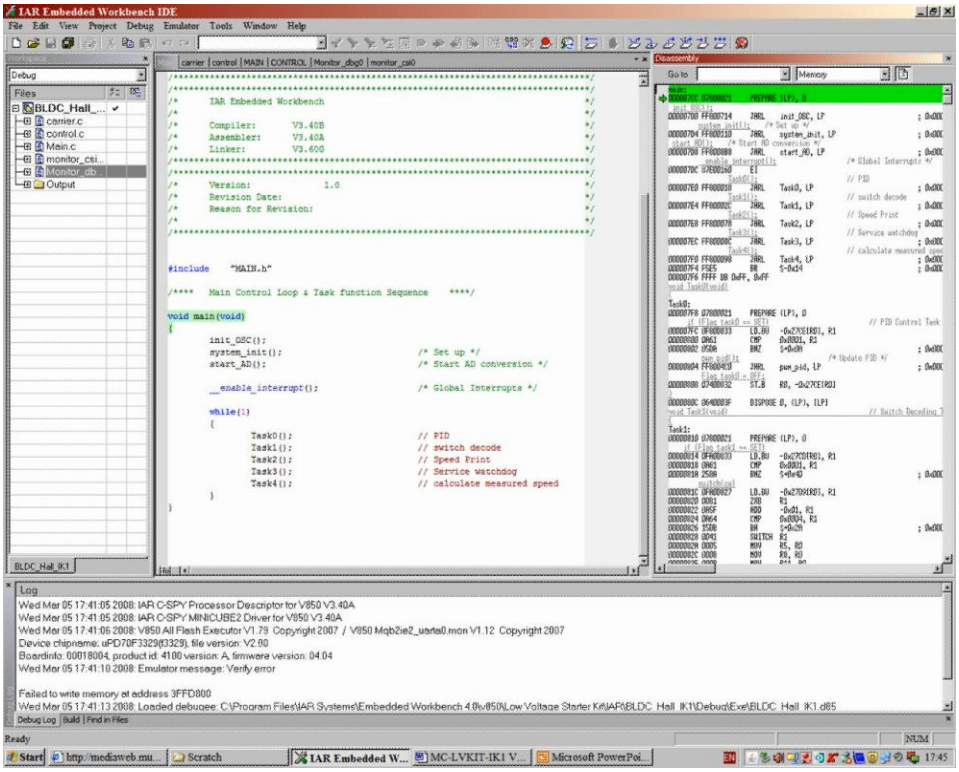

Figure 8-31 Integrated Debugger - Main Window

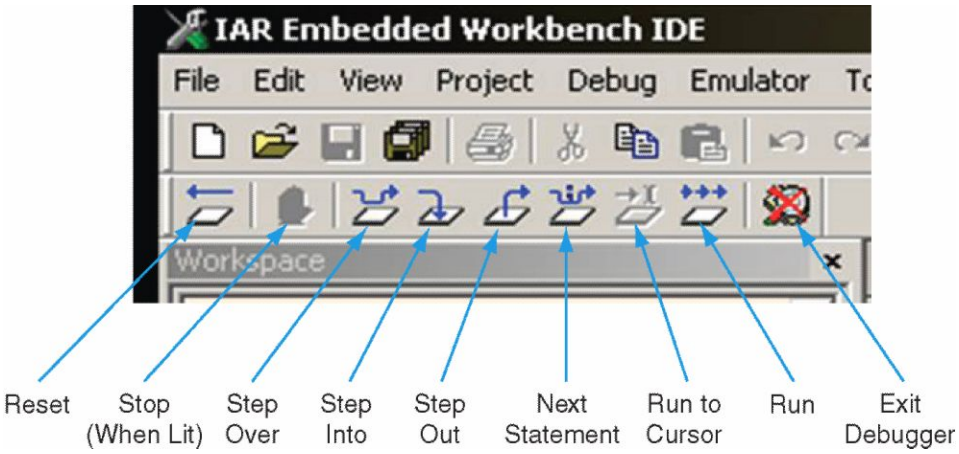

Figure 8-32 Debugger Task Bar Icons

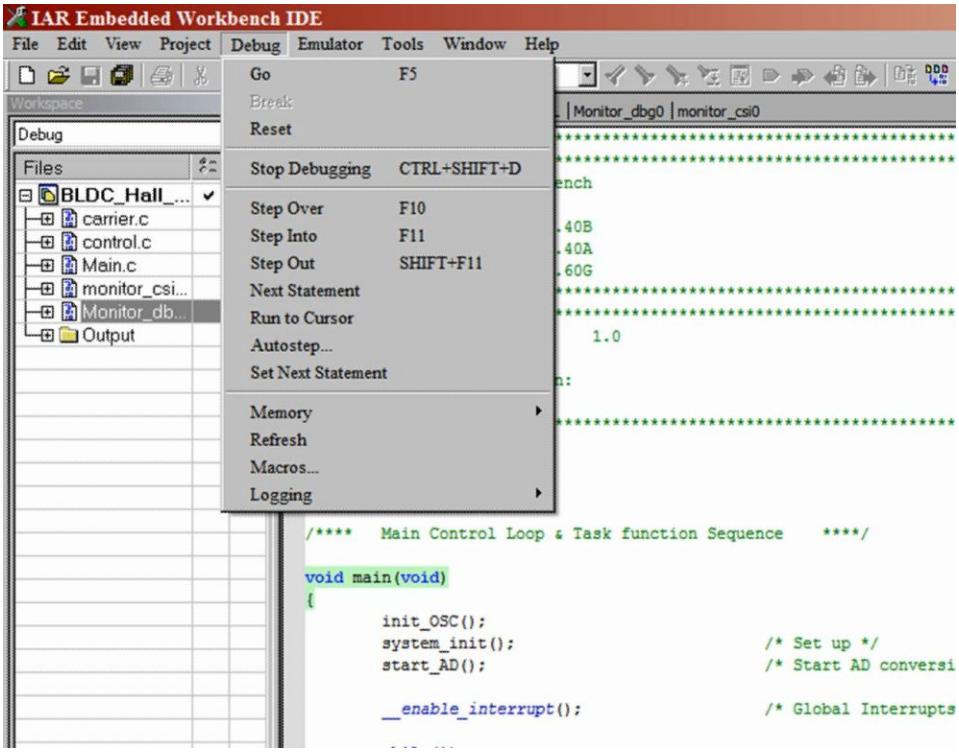

Figure 8-33 Debug Menus

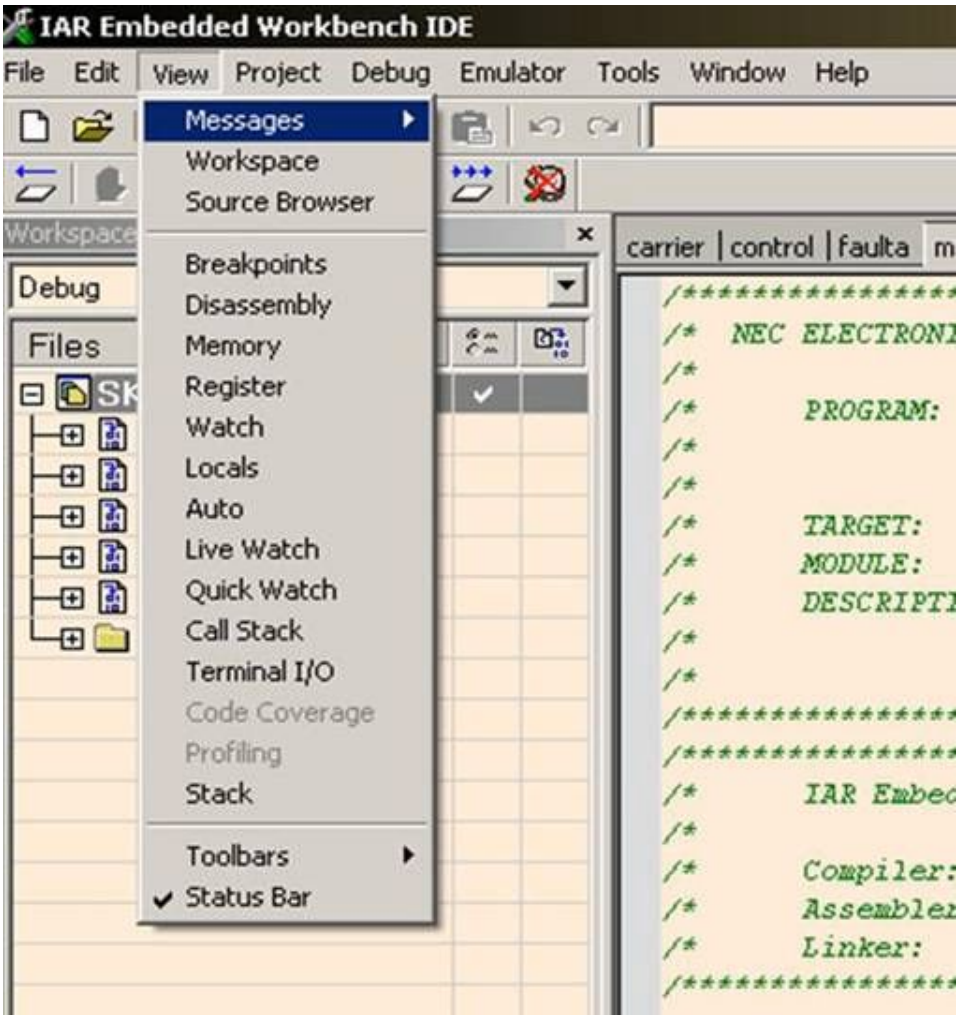

Figure 8-34 Debug Views Windows

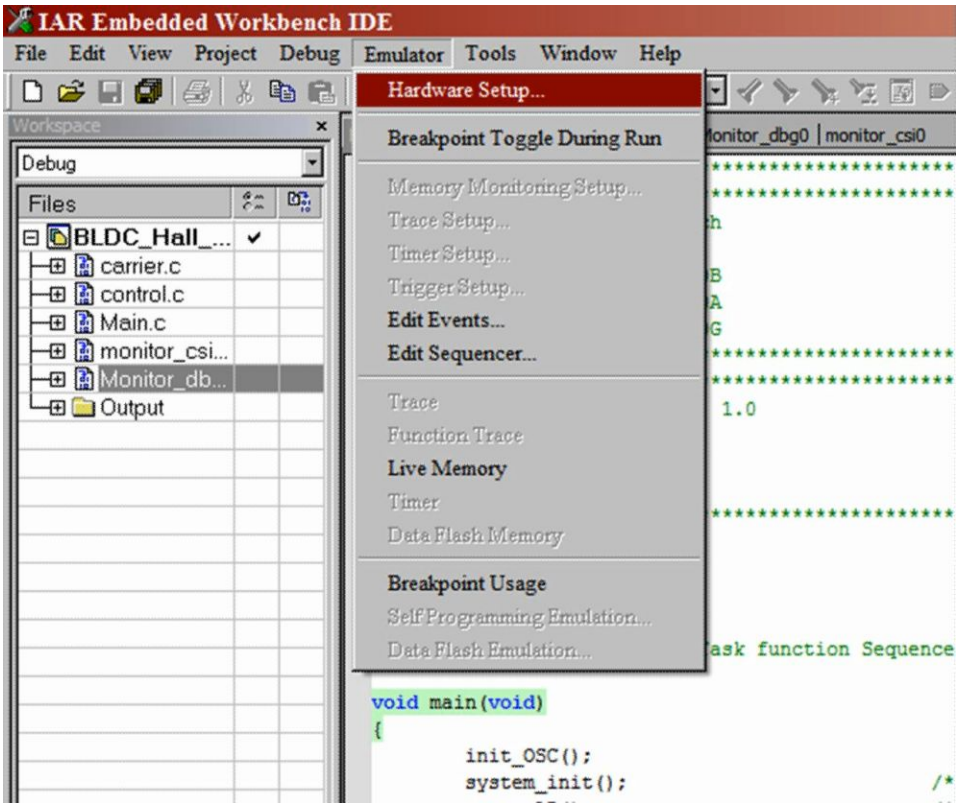

Figure 8-35 Emulator Debug Options

Note The "Live Watch" does not operate in real time on the on-chip debug unit.